



NRL/MR/5620--99-8332

# Spectral Calibration, Spatial Mapping and Flat Fielding Studies of the Dark HORSE 1 (DH1) March Data Collection

CHRISTOPHER M. STELLMAN  
GEOFFREY G. HAZEL  
JONATHON M. SCHULER  
FRANK BUCHOLTZ  
JOSEPH V. MICHALOWICZ

*Advanced Concepts Branch  
Optical Sciences Division*

January 29, 1999

19990127 018

Approved for public release; distribution unlimited.

| REPORT DOCUMENTATION PAGE  |  |   | Form Approved<br>OMB No. 0704-0188                                   |  |
|--|--|---|--|--|
| Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.   |  |   |  |  |
| 1. AGENCY USE ONLY (Leave Blank)   | 2. REPORT DATE<br><br>January 29, 1999                       | 3. REPORT TYPE AND DATES COVERED<br><br>Final Report        |  |  |
| 4. TITLE AND SUBTITLE<br><br>Spectral Calibration, Spatial Mapping and Flat Fielding Studies of the Dark HORSE (DH1) March Data Collection   |  |   | 5. FUNDING NUMBERS   |  |
| 6. AUTHOR(S)<br><br>Christopher M. Stellman, Geoffrey G. Hazel, Jonathan M. Schuler, Frank Bucholtz, and Joseph V. Michalowicz   |  |   |  |  |
| 7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)<br><br>Naval Research Laboratory<br>Washington, DC 20375-5320   |  |   | 8. PERFORMING ORGANIZATION REPORT NUMBER<br><br>NRL/MR/5620--99-8332 |  |
| 9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)<br><br>Office of Naval Research<br>Arlington, VA 22217-5320  |  |   | 10. SPONSORING/MONITORING AGENCY REPORT NUMBER                       |  |
| 11. SUPPLEMENTARY NOTES  |  |   |  |  |
| 12a. DISTRIBUTION/AVAILABILITY STATEMENT<br><br>Approved for public release; distribution unlimited.   |  |   | 12b. DISTRIBUTION CODE<br><br>A                                      |  |
| 13. ABSTRACT (Maximum 200 words)<br><br>Recently researchers at the Naval Research Laboratory (NRL) have successfully demonstrated autonomous, real time, hyperspectral detection of airborne and military ground targets. Real-time autonomous cueing of a high-resolution imager, and designation of targets with pointing optics and a pulser laser was also demonstrated. The work was performed under NRL's Dark HORSE effort, the culmination of a four-year Multispectral Overhead IR/EO Surveillance (MOVIES) program to develop real-time hyperspectral detection, cueing, target location, and target designation capabilities. The program will provide enabling technology for future manned, reconnaissance, and Uninhabited Combat Air Vehicle (UCAV) systems. The following describes the spectral calibration and flat fielding studies performed on a recently collected data set of the Dark HORSE 1 (DH1) visible hyperspectral sensor. The procedures used for spectral calibration and mapping of the sensor's focal plane array (FPA) are described and a real-time algorithm for corrective re-mapping of the image data is presented in detail. Also, procedures used for pre and post-flight flat fielding of the sensor's radiometric response are described with an emphasis on the potential advantages and disadvantages of each. Results of these efforts are discussed and appropriate conclusions are drawn. |  |   |  |  |
| 14. SUBJECT TERMS<br><br>Hyperspectral detection      Spectral calibration method<br>Flat fielding methods      Remote sensing   |  |   | 15. NUMBER OF PAGES<br><br>47  |  |
|  |  |   | 16. PRICE CODE   |  |
| 17. SECURITY CLASSIFICATION OF REPORT<br><br>UNCLASSIFIED  | 18. SECURITY CLASSIFICATION OF THIS PAGE<br><br>UNCLASSIFIED | 19. SECURITY CLASSIFICATION OF ABSTRACT<br><br>UNCLASSIFIED | 20. LIMITATION OF ABSTRACT<br><br>UL                                 |  |

## **CONTENTS**

|     |  |    |
|-----|--|----|
| 1.  | INTRODUCTION   | 1  |
| 2.  | DATA COLLECTION INSTRUMENT (Dark HORSE 1)            | 2  |
| 3.  | FIELD DATA   | 10 |
| 4.  | SPECTRAL AND SPATIAL RE-SAMPLING AND CALIBRATION     | 12 |
| 5.  | GENERATION OF PRE-FLIGHT FLAT FIELDING COEFFICIENTS  | 15 |
| 6.  | REAL-TIME IN-FLIGHT CORRECTIONS                      | 17 |
| 7.  | GENERATION OF POST-FLIGHT FLAT FIELDING COEFFICIENTS | 18 |
| 8.  | RESULTS AND DISCUSSION                               | 18 |
|     | Spectral Re-sampling and Calibration                 | 18 |
|     | Pre-Flight Flat Fielding                             | 19 |
|     | Post-Flight Flat Fielding                            | 24 |
|     | Real-time Anomaly Detection                          | 27 |
| 9.  | REFERENCES   | 31 |
| 10. | APPENDIX   | 32 |

# **SPECTRAL CALIBRATION, SPATIAL MAPPING AND FLAT FIELDING STUDIES OF THE DARK HORSE 1 (DH1) MARCH DATA COLLECTION**

## **1. INTRODUCTION**

The need for strategic military surveillance of critical airborne and ground targets has been well recognized in the military community. Multispectral and hyperspectral sensors offer the possibility of exploiting the spectral differences between targets of interest and local backgrounds as a detection discriminant. In addition, rapid developments in computing power and anomaly detection algorithms have led to the possibility of real-time target identification, location, and designation. Recently researchers at the Naval Research Laboratory (NRL) have successfully demonstrated autonomous, real time, hyperspectral detection of airborne and military ground targets [1]. Real-time autonomous cueing of a high-resolution imager, and designation of targets with pointing optics and a pulsed laser was also demonstrated. The work was performed under NRL's Dark HORSE effort, the culmination of a four-year Multispectral Overhead IR/EO Surveillance (MOVIES) program to develop real-time hyperspectral detection, cueing, target location, and target designation capabilities [2]. The program will provide enabling technology for future manned, reconnaissance, and Uninhabited Combat Air Vehicle (UCAV) systems.

The following describes the spectral calibration and flat fielding studies performed on a recently collected data set of the Dark HORSE 1 (DHI) visible hyperspectral sensor. The procedures used for spectral calibration and mapping of the sensor's focal plane array (FPA) are described and a real-time algorithm for corrective re-mapping of the image data is presented in detail. Also, procedures used for pre and post-flight flat fielding of the sensor's radiometric response are described with an emphasis on the potential advantages and disadvantages of each. Results of these efforts are discussed and appropriate conclusions are drawn.

## 2. DATA COLLECTION INSTRUMENT (Dark HORSE 1)

The complete DH1 system is composed of eight components; a visible hyperspectral sensor, a sensor controller, a real-time processor, a system controller, a high resolution camera, a ball gimbal laser designator, an air-to-ground RF data link and an infrared camera. Figure 1 is a block diagram of the entire system, except the data link, and outlines the typical operation of the DH1 system. Data from the hyperspectral sensor are analyzed by the real-time processor and are shown as a continuous waterfall display (lower left). When a target is detected, the high-resolution camera is instructed to take a picture (center) and the laser is pointed at the target. Both ground targets and an aircraft are shown. Images and target information are transmitted to the ground via the RF Common Data Link.

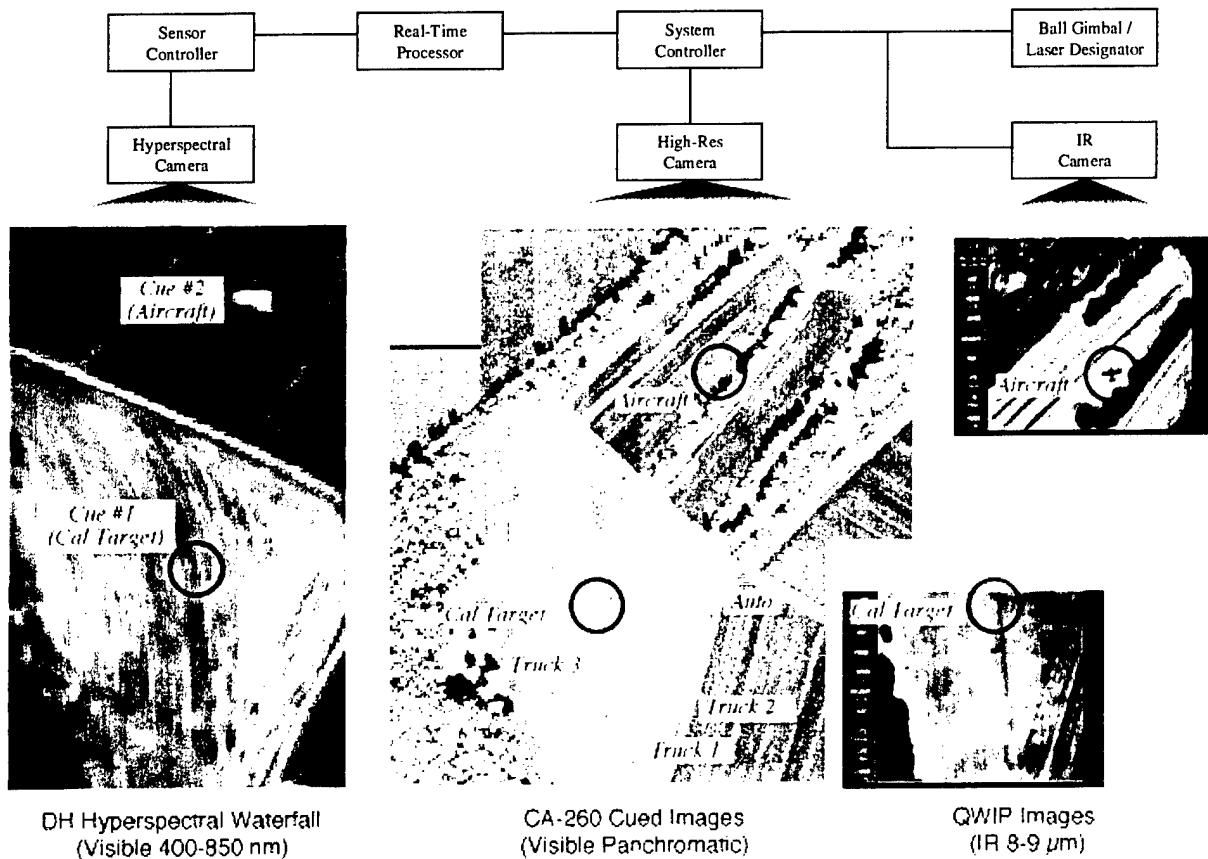
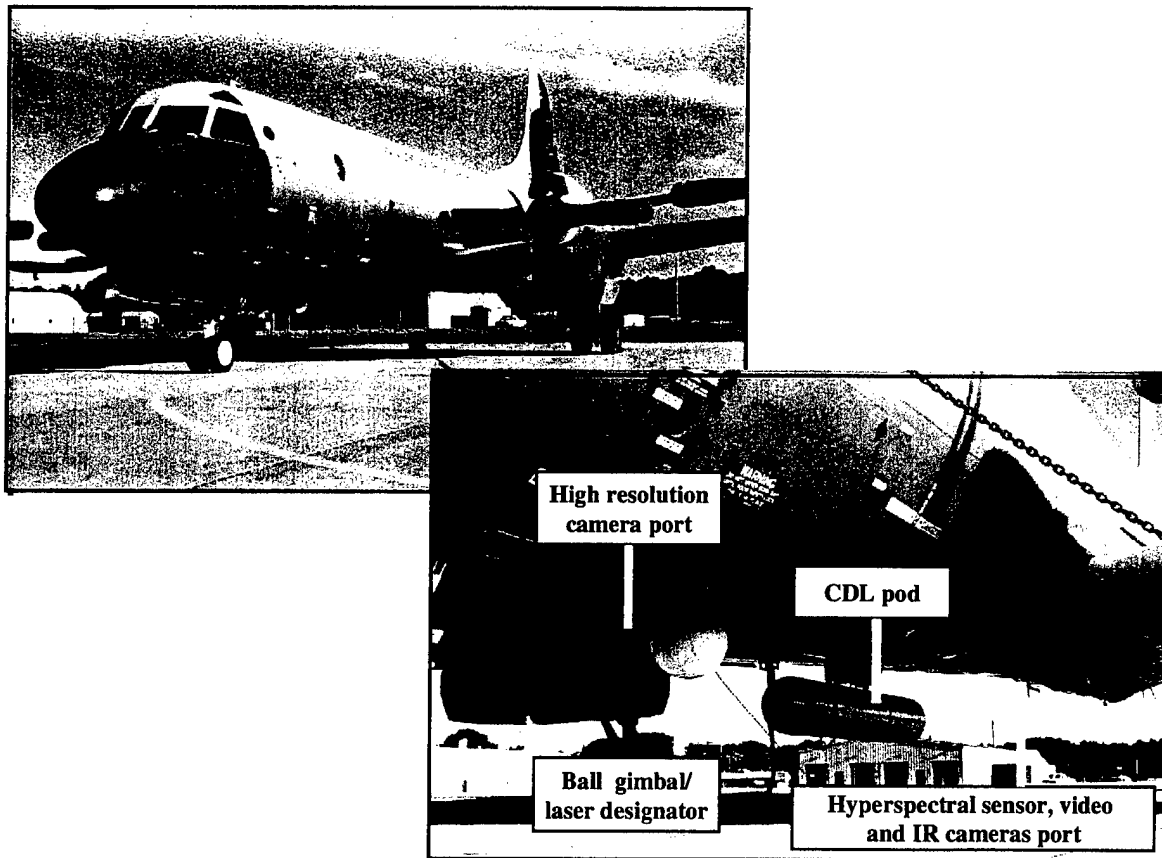


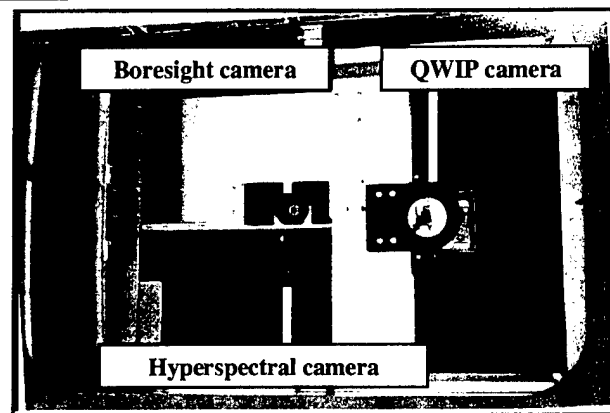
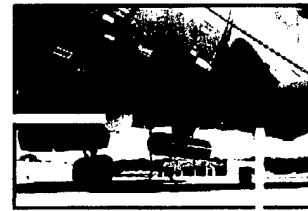
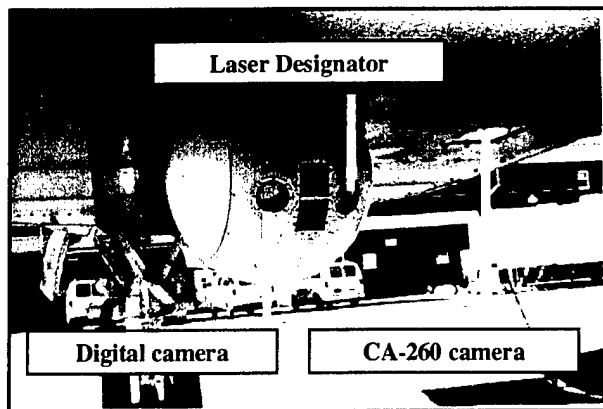
Figure 1: Components and Operation of the DH1 System.

Figure 2 and Figure 3 show the complete system installed aboard a P-3 aircraft provided by NRL's Flight Support Detachment, Patuxent River Naval Air Station, MD. Detection algorithms have been developed by NRL with the support of SCC, Inc., Los Angeles, CA. Cued high-resolution imagery has been obtained using NRL's ROI CA-260 25 megapixel camera and data has been transmitted to a ground station at NRL, Washington, DC, using an Advanced Tactical Airborne Reconnaissance System Common Data Link (ATARS-CDL).

The work described in this document focuses on the spectral calibration, spatial mapping and flat fielding of the DH1 visible hyperspectral sensor and the affects of these processes on collected data. Therefore, only the sensor, the sensor controller and the real-time processor are described in detail.



*Figure 2: Dark HORSE 1 System Installed on the NRL P-3 Aircraft.*



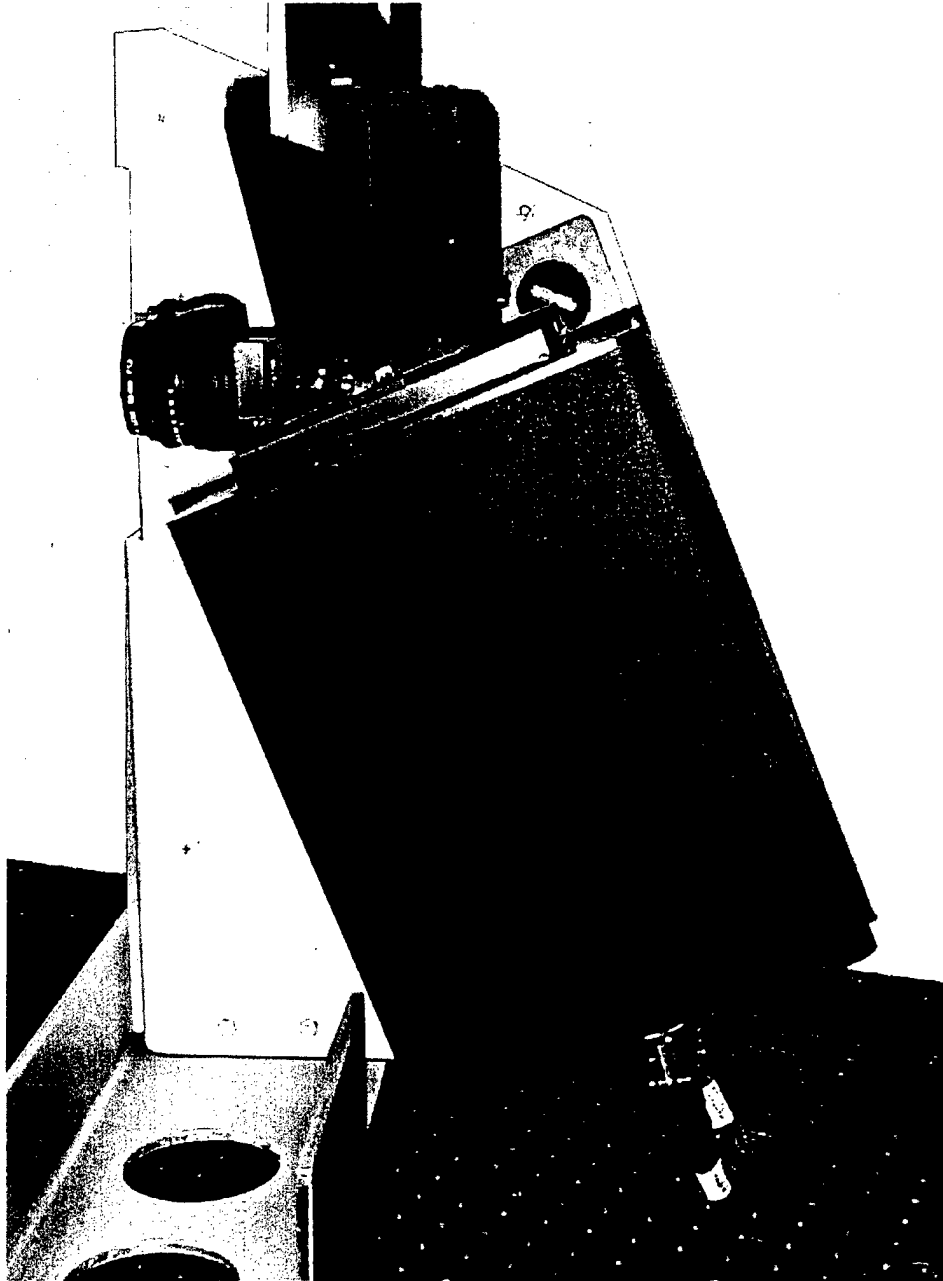
*Figure 3: Close-up of Dark HORSE 1 System Installed on the NRL P-3 Aircraft.*

The hyperspectral imaging sensor (Figure 4) consists of a standard 1" CCTV lens (Navitar, DO-5018), a  $f/2$  imaging spectrograph (Instruments S.A., CP140) and a high-frame-rate CCD camera (Sarnoff, VCCD512).

Imagery collected via the 50mm lens is dispersed using a high throughput  $f/2$  imaging spectrograph. The spectrograph employs an aberration-corrected concave holographic grating of custom design, providing a "flat" field spectral range from 400 to 850 nm. For all collected data a 50  $\mu\text{m}$  slit width was used. However, it should be noted that the size of the binned CCD pixels (not the spectrograph slit width) defines the sensors maximum spectral resolution of 7nm.

Digital collection of the hyperspectral imagery is achieved using a custom high-frame-rate CCD camera. The camera is a 16-port split frame transfer CCD with 12-bit digitizers and operates at a maximum frame rate of 200 Hz (100 Hz when operated in conjunction with the real-time processor and algorithms). The 512 x 512 silicon FPA is capable of capturing digital hyperspectral data cubes with a maximum of 128

cross-track spatial pixels and 64 wavelength bands. A custom high-frame-rate interface box is used to merge the 16 12-bit digital camera output channels into a single 32-bit channel that is then read via a digital frame grabber board (MuTech, MV-1100).



*Figure 4: The DH1 Hyperspectral Imaging Sensor.*



The sensor is controlled using a 266 MHz Pentium II PC and a custom software application run under the Microsoft Windows NT operating system. The user interface (Figure 5) provides a method of defining which DH computers are brought online and gives the user a means of entering required input parameters; such as archival file names, correction coefficient files and camera frame rates. The interface also provides the user a mechanism for simultaneous starting and stopping of the DH1 sensor and supporting anomaly detection algorithms (located in the real time processor).

**PC1 Control**

**STATUS**

PC1: Ready

PHILLS: OK

Processing: OK

PC2 Link: Disabled.

PC3 Link: Disabled.

☒ Archive: C:\Example

Frame Rate: 50

☐ Fake Data

☐ Disable Processing

☒ Disable PC2 Link

☐ Enable Processed Archive

☒ Disable PC3 Link

Initialize

Quit

Run

Stop

**Project DARK HORSE**  
Naval Research Laboratory

Figure 5: DH1 Sensor Controller User Interface.

Two anomaly detection algorithms are run in parallel using a real-time processor. Both the subspace R-X [3] and LBG clustering [4] algorithm employ spatial filtering. The real-time processor consists of a 200 MHz Pentium Pro with a DSP subsystem (8 Analog Devices SHARC) and provides greater than 1 Gflop/s of peak processing power. A custom software application run under the Microsoft Window NT operating system provides the user a means of changing algorithm parameters and thresholds, and gives the user data output in the form of a real-time waterfall display (Figure 6).

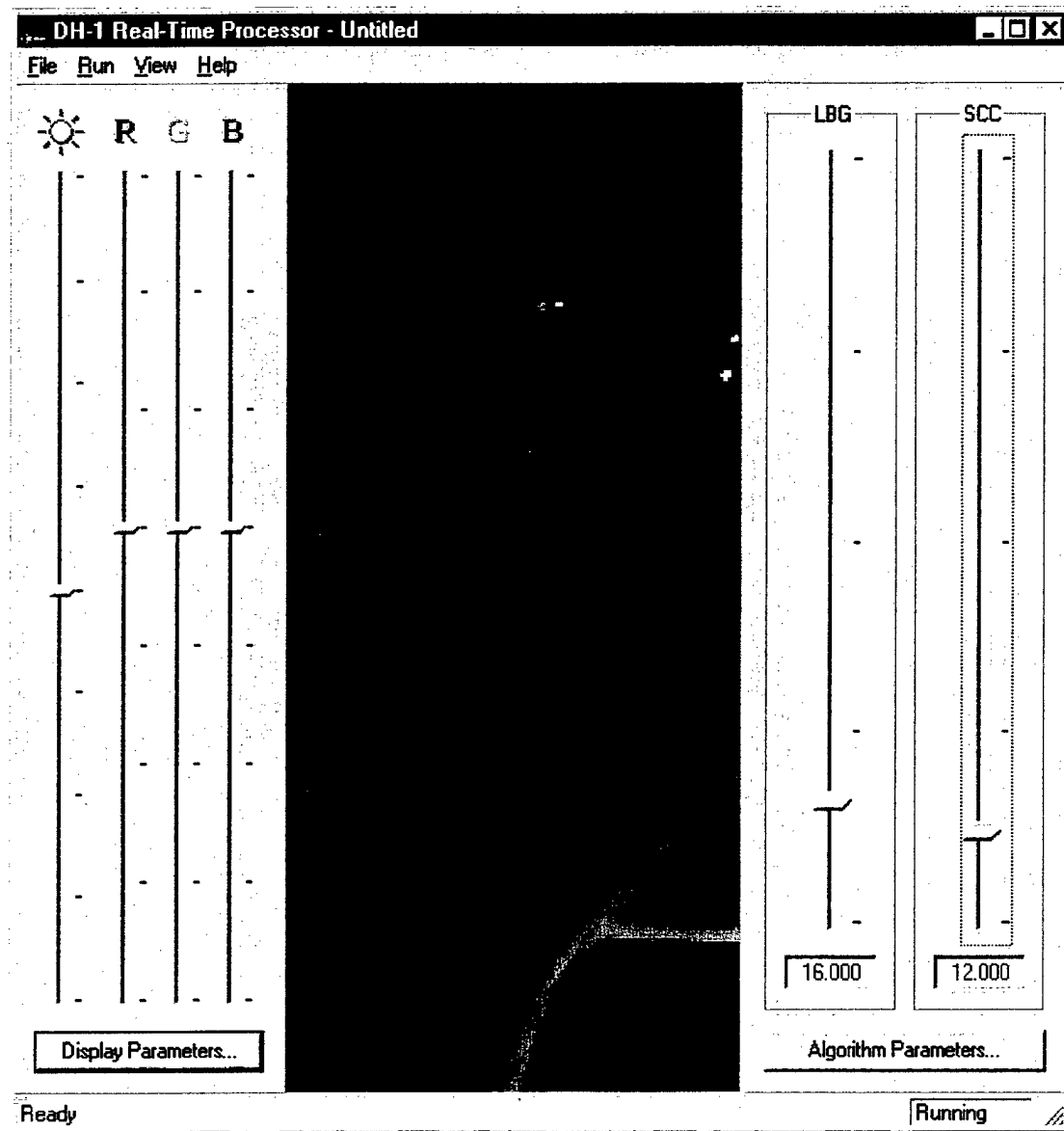


Figure 6: DH1 Real-time Processor User Interface.

For the subspace R-X algorithm the user can define the wavelength bands used for spectral analysis, the number of lines used for recursive filtering and the PC bands used for R-X data modeling (Figure 7). For the LBG clustering algorithm the user can define the wavelength bands used for spectral analysis, the number of clusters employed, the number of lines used for initialization and decimation, and the number of clustering iterations used during initialization and operation (Figure 7).

Algorithm Parameters

LBG

SCC

Use spatial filtering

☒

Effective number of lines averaged in recursive filters:

10

Retain PC bands

2

thru

14

Bands:

|   |    |    |    |    |    |    |    |
|---|----|----|----|----|----|----|----|
| 0 | 8  | 16 | 24 | 32 | 40 | 48 | 56 |
| 1 | 9  | 17 | 25 | 33 | 41 | 49 | 57 |
| 2 | 10 | 18 | 26 | 34 | 42 | 50 | 58 |
| 3 | 11 | 19 | 27 | 35 | 43 | 51 | 59 |
| 4 | 12 | 20 | 28 | 36 | 44 | 52 | 60 |
| 5 | 13 | 21 | 29 | 37 | 45 | 53 | 61 |
| 6 | 14 | 22 | 30 | 38 | 46 | 54 | 62 |
| 7 | 15 | 23 | 31 | 39 | 47 | 55 | 63 |

Apply

Close

Algorithm Parameters

LBG

SCC

Use spatial filtering

☒

Number of clusters:

8

Number of lines used at initialization:

8

Line decimation at initialization:

1

Number of clustering iterations during initialization:

10

operation:

2

Bands:

|   |    |    |    |    |    |    |    |
|---|----|----|----|----|----|----|----|
| 0 | 8  | 16 | 24 | 32 | 40 | 48 | 56 |
| 1 | 9  | 17 | 25 | 33 | 41 | 49 | 57 |
| 2 | 10 | 18 | 26 | 34 | 42 | 50 | 58 |
| 3 | 11 | 19 | 27 | 35 | 43 | 51 | 59 |
| 4 | 12 | 20 | 28 | 36 | 44 | 52 | 60 |
| 5 | 13 | 21 | 29 | 37 | 45 | 53 | 61 |
| 6 | 14 | 22 | 30 | 38 | 46 | 54 | 62 |
| 7 | 15 | 23 | 31 | 39 | 47 | 55 | 63 |

Apply

Close

Figure 7: DHI Real-time Processor Parameter Listing User Interface.

The following table provides an overview of system specifications for the hyperspectral sensor and controller, real-time processor and high-resolution camera. Some typical flight parameters are also listed.

| <b>Hyperspectral Sensor and Controller</b><br>(Developed by NRL) |  |
|--|--|
| Spectrometer Type  | Grating, 50 $\mu$ m Slit                 |
| Wavelength Range   | 400 – 850nm                              |
| CCD Array  | 512 x 512 Silicon                        |
| Frame Rate   | 25 – 200 Hz (50 Hz Typical)              |
| Pixel Depth  | 12 bit                                   |
| Output Data Size   | 128 Pixels x 64 Bands x 32 bit Precision |

| <b>Real-Time Processor</b><br>(Developed by SCC and NRL) |                                       |
|--|---------------------------------------|
| Platform   | Pentium Pro, 200 MHz                  |
| DSP Processor  | 8 Analog Devices SHARC                |
| Peak Processor Power                                     | > 1Gflop/s                            |
| Algorithm #1   | Subspace R-X with Spatial Filtering   |
| Algorithm #2   | LBG Clustering with Spatial Filtering |

| <b>High-Resolution Camera</b><br>(Developed by ROI) |                      |
|---|----------------------|
| Camera Type   | CA 261               |
| Wavelength Range                                    | Visible Panchromatic |
| CCD Array   | 5M x 5M              |

| <b>Typical Flight Parameters</b>                                   |   |
|--|---|
| Altitude   | 6500 ft                                 |
| Air Speed  | 100 m/s                                 |
| Hyperspectral Ground Pixel Size<br>(50mm Lens, 50 Hz, 2:1 Binning) | Down-track: 3.9m<br>Cross-track: 1.2m   |
| High Resolution Ground Pixel Size<br>(ROI Camera)                  | Down-track: 0.08m<br>Cross-track: 0.08m |

### 3. FIELD DATA

The field data set, to which the calibration and flat fielding were applied, was collected over the Army Research Laboratory (ARL) site at Blossom Point, MD (Lat. N 38.41009014, Long. W 77.10263469) on March 5, 1998. A map of the region showing the relative location of the ground targets and approximate flight path of the aircraft is shown in Figure 8. The targets consisted of an 8' x 8' gray wooden platform and an 8' x 12' brown camouflage tarp. Respectively, the targets were chosen to have significantly different and similar spectral signatures to that of the surrounding field (as seen by the human eye).

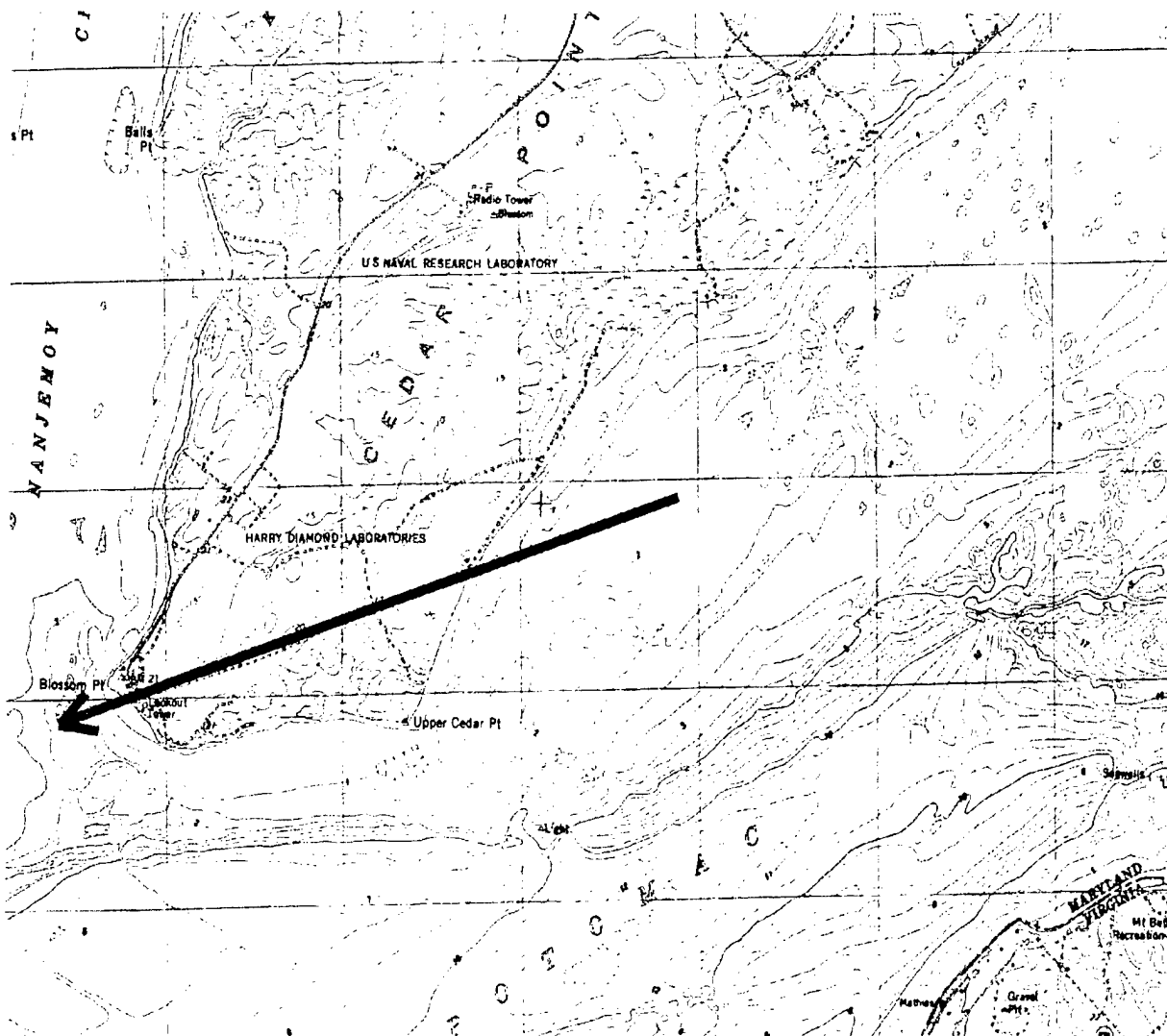
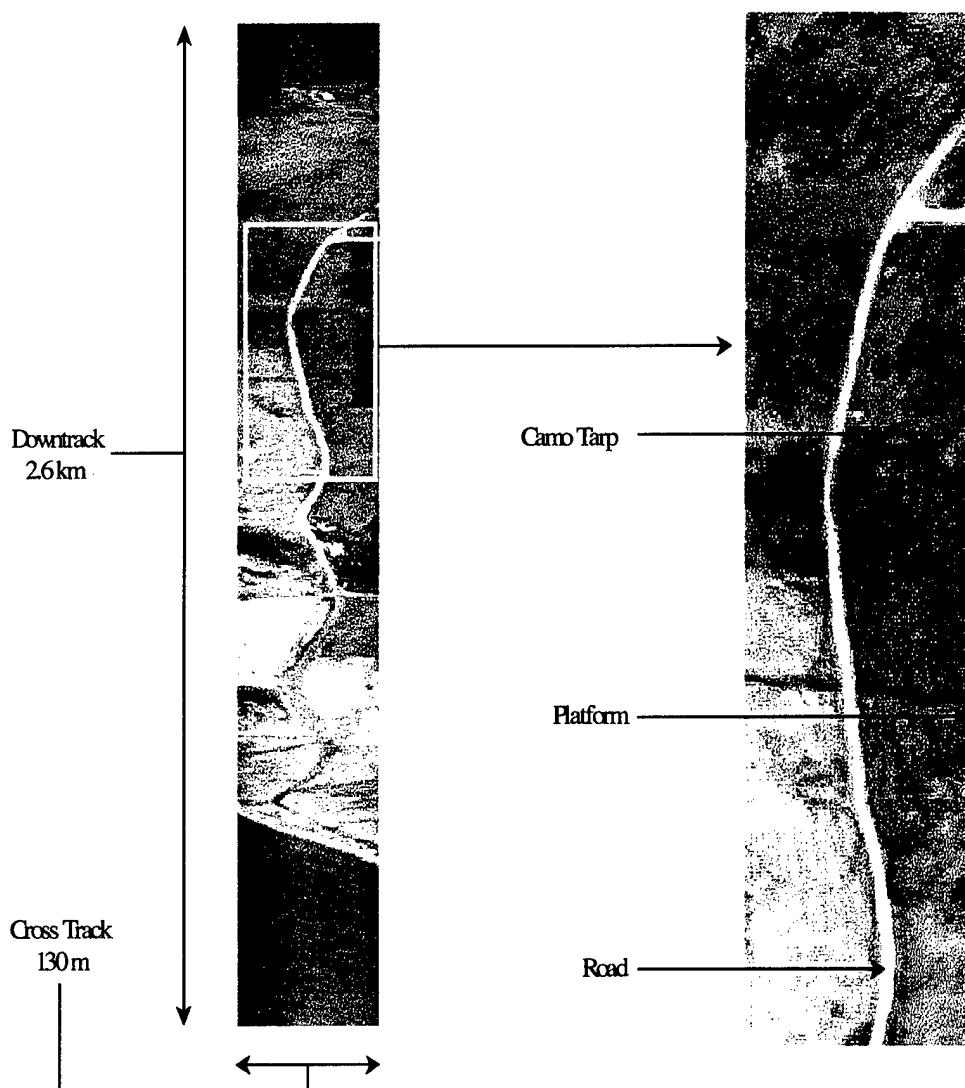


Figure 8: Target Region, ARL, Blossom Point, MD.

Field data was collected by flying an east to west heading over the target region at an altitude of 6500 feet and an air speed of 100m/s. For this study the sensor employed a 50 mm lens operated at f/5.6 and was operated at 50 Hz. This provided a 50% overlap of consecutive frames and resulted in a hyperspectral cross-track ground pixel size of 1.2 m and a down-track ground pixel size of 3.9 m. Data was collected over a 10 km path to include sampling of water regions both prior to and after the target region of interest. Weather was partly cloudy at 7000 feet and strong crosswinds accounted for considerable aircraft roll.



*Figure 9: Hyperspectral Image of the Target Region, ARL, Blossom Point, MD.*

Nineteen data sets were collected over the target site. These measurements include observations of several combinations of the targets being present or removed from the region of interest, as well as the use of both blue and brown camouflaged tarps. For the purposes of this study only data set number 12 has been employed. This data set was chosen because both the platform and camouflage targets were present and minimal roll was encountered during the course of the run. The data set was collected at 2:01 PM and consists of 5144 consecutive frames. Figure 9 shows a section of the data set over the land region of interest (frames 3400-4700).

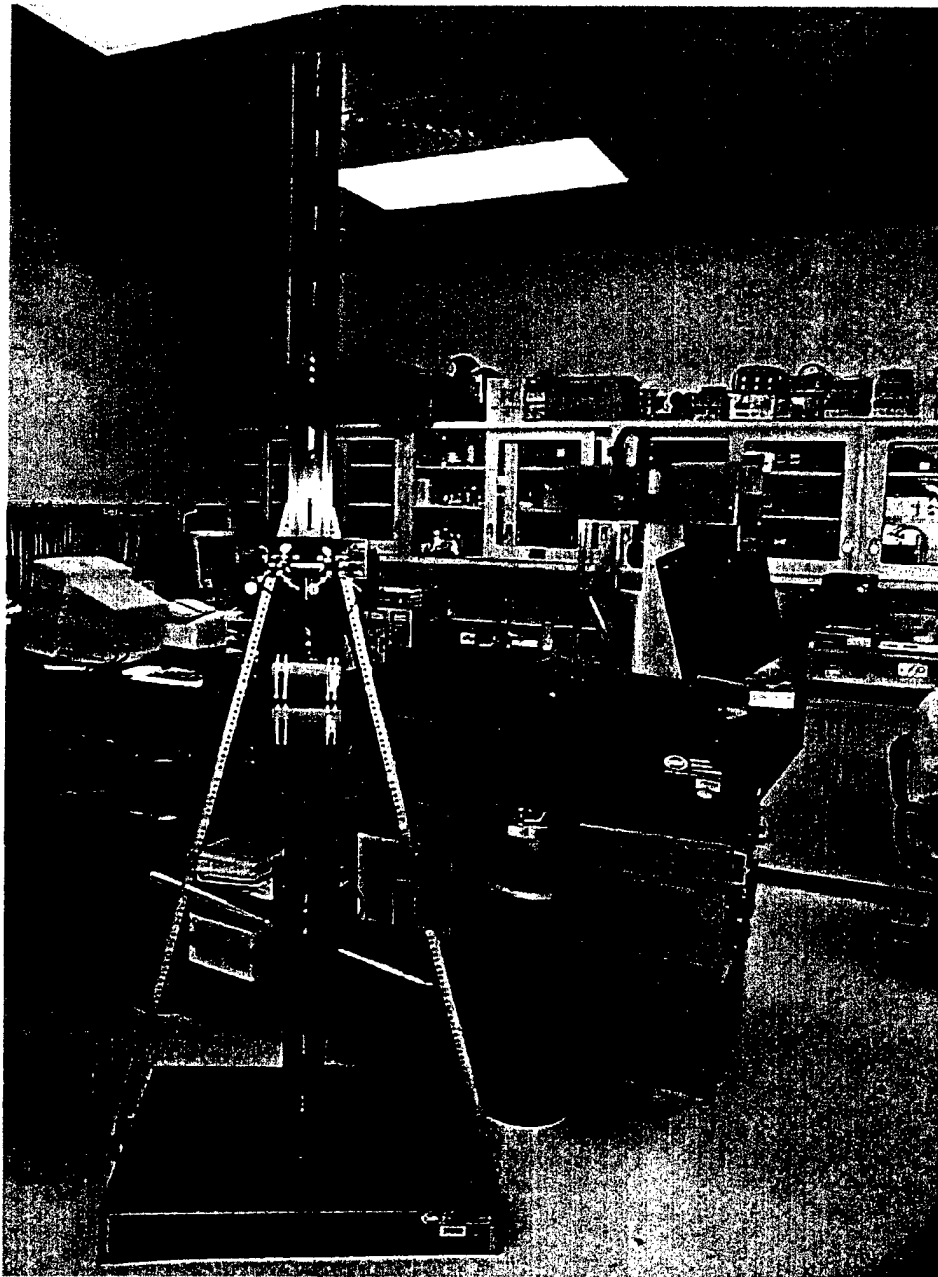
#### **4. SPECTRAL AND SPATIAL RE-SAMPLING AND CALIBRATION**

Prior to flight the DH1 sensor was spectrally calibrated to ensure proper wavelength assignments could be made to the 64 collected wavelength bands. Unfortunately, this process is not as straightforward as one might expect. As previously stated the DH1 sensor employs an aberration-corrected holographic grating. However, due to aberration contributions from the front end optics and the limitations of trying to correct for such aberrations via a single optical grating element, the sensor is still plagued with a significant degree of image distortion at the sensor focal plane (CCD). The distortion is that of a classic pin cushion type and is consistent with a curvature of field and/or spherical aberration. Because the use of multiple optical elements for aberration correction is not a feasible option (due to instrumental size, time and cost constraints) a real time software correction is used to eliminate image distortion. The correction is done prior to running the parallel anomaly detection algorithms and relies on spectral and spatial re-mapping of the collected imagery. Re-mapping is achieved by re-sampling the FPA data via software.

All pre-flight spectral and spatial mapping data was acquired in room 1036 of building 216, NRL. The instrumentation used is shown in Figure 10. The custom mounting rack is used to hold two "pencil style" spectral calibration lamps (Oriel, 6035 and 6031) and their supporting power supply units. The Mercury and Krypton lamps are used simultaneously and provide numerous discrete spectral lines over the sensor's 400 to 850 nm wavelength range.

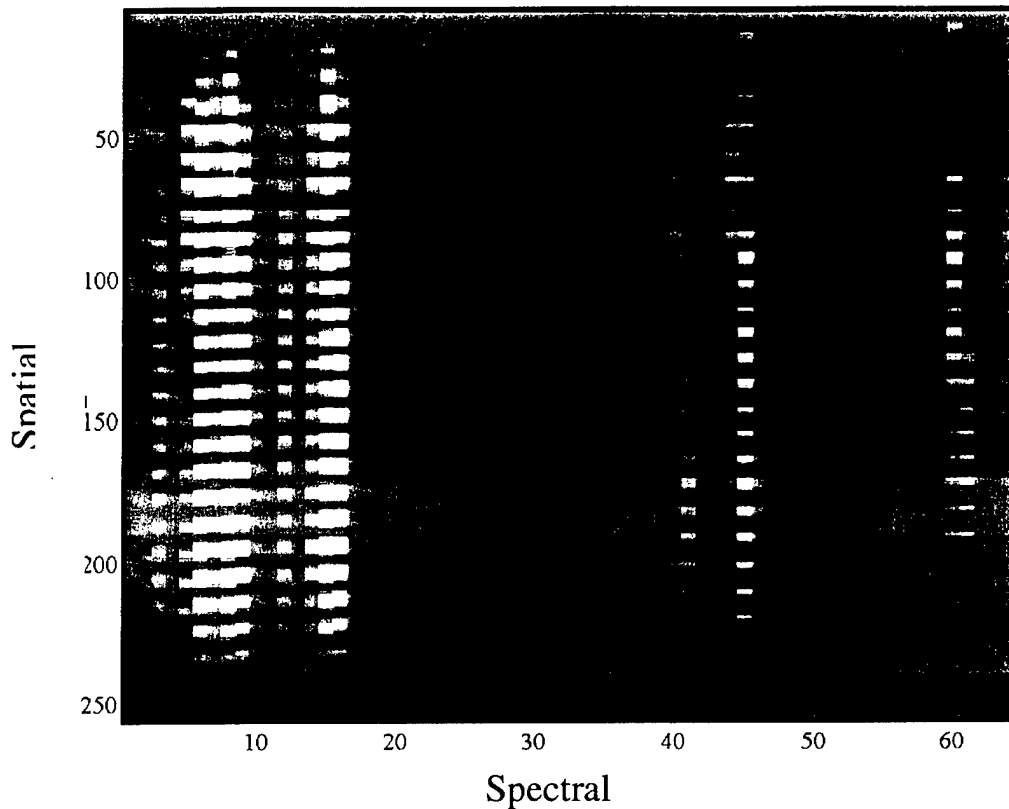
A map of the sensor's spectral and spatial distortion was compiled (Figure 11) by scanning the spectral lamps to known vertical positions across the sensor's field of view and collecting image frames at each location. The camera frame rate was adjusted at each vertical location to keep spectral line intensities near the middle of the sensors dynamic range. An offset correction was performed on the calibration lamp data by dark frame subtraction. Dark frames were collected at each frame rate used in the lamp data collection by installing the camera lens cap. In some instances an appropriate dark frame was missing or unusable for a particular calibration lamp frame. In these instances another lamp frame from a distant

vertical position was used as a dark frame. This provided some local offset correction near the location of the lamp image. The corrected data was then combined into a single frame by cropping the lamp image from each lamp frame and reassembling the cropped images into a collage frame for peak finding. The resulting frame had an irregular grid of bright spots corresponding to the spectral peaks of the lamps at each spatial location. These spots needed to be located by the peak finding routine.



*Figure 10: Instrumentation used for Spectral and Spatial Re-Mapping of the DHI Sensor.*





*Figure 11: Spectral and Spatial Map of the DH1 Sensor.*

The automated peak finding routine divided the frame into windows containing a few peaks each. All local maxima were taken as initial peak location estimates. A two-dimensional Gaussian surface was used to model each peak. A separate DC offset for each FPA sector was included in the model. A nonlinear optimization procedure was performed to adjust the Gaussian means and variances and the DC offsets to fit the window data. A simple gradient descent technique was employed for optimization.

The automated peak finding routine was error prone. There were both spurious peaks found and real peaks lost. In addition, many of the real peaks were split or otherwise distorted by residual non-uniformity in the array. Therefore, manual editing of the peak locations was performed until a reasonable match to the clearly discernable peaks in the image was obtained.

Next, the grid of peaks was regularized. The center row of peaks corresponding to the central spatial position of the calibration lamps was chosen as the spectral standard. All peaks corresponding to the same spectral line of the lamps were aligned with the spectral standard. This removed the spectral misregistration (i.e. all light of the same wavelength would appear in the same spectral column regardless of

its spatial origin in the camera's far field ). Next a centrally located column of peaks corresponding to the centermost spectral line of the lamps was chosen as a spatial standard. This standard was then corrected to be uniformly spaced across the field of view of the camera. This achieved a spatial calibration of the camera. All peaks corresponding to the same spatial location of the lamps were then aligned with the corrected spatial standard. This removed the spatial misregistration (i.e. all light from the same point in the camera's far field would appear in the same spatial row regardless of its wavelength).

The result of these steps was an offset vector for each of the identified peaks in the calibration lamp frame. This can be viewed as a non-uniform grid of 'delta' vectors. The full coordinate transformation for the entire focal plane grid (a 'delta' vector for each pixel) was derived by a two-dimensional biharmonic spline interpolation of the non-uniform grid onto the full focal plane grid. The 'delta' vectors were then reinterpreted to give for each pixel in the regularized image a corresponding (x,y) location in the focal plane. With this information new FPA data can be regularized by performing an interpolation on the FPA pixels in the neighborhood of the (x,y) location corresponding to each pixel in the regularized image. For example, if pixel (i,j) in the regularized image corresponds to position (x,y) in the focal plane, then the value of pixel (i,j) can be obtained from a bilinear interpolation of the four FPA pixels closest to position (x,y).

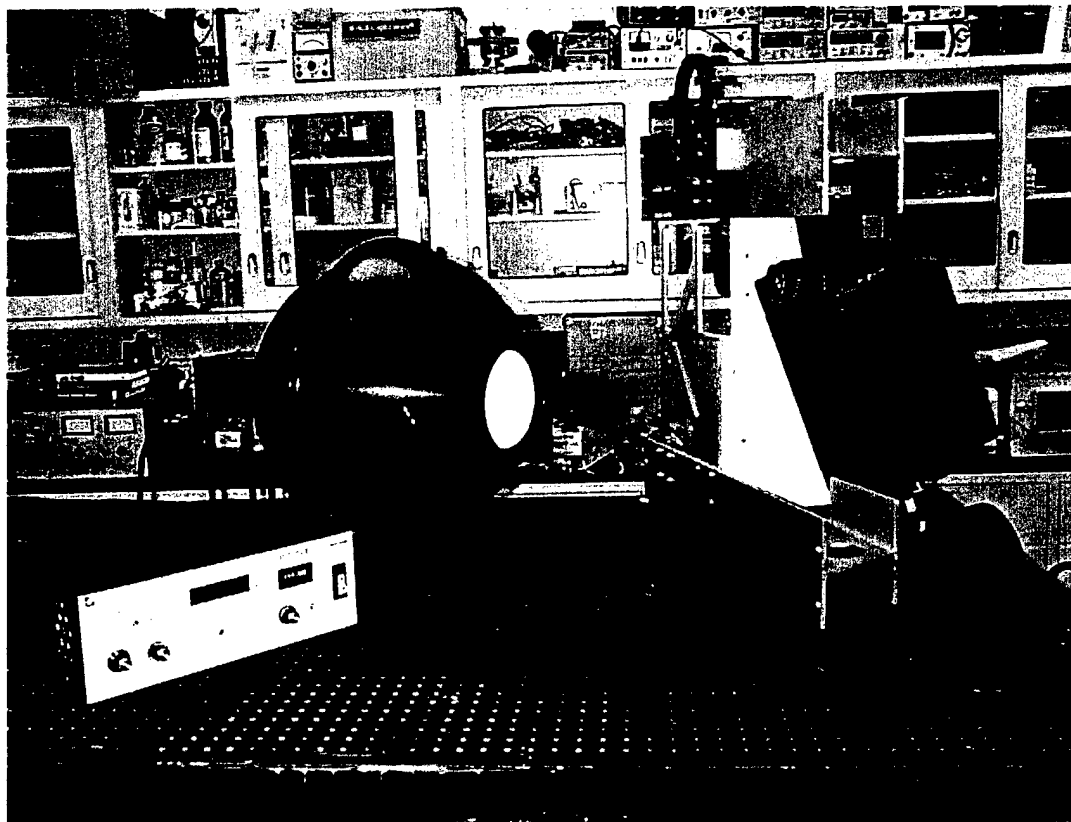
Once the regularized data was obtained, a spectral calibration of the camera could be performed. Recall that the spatial calibration was performed together with the regularization. The first step in the calibration was a precise identification of the spectral lines visible in the data. This was achieved by comparing published spectral line tables with a spectrum obtained from a regularized image of the spectral lamps taken with the lamps in the near field of the camera lens. Once a wavelength was assigned to each observed peak, the center wavelength of each of the spectral columns of the regularized imagery was estimated from a least squares fit to a linear dispersion model. That is, a linear least square fit was performed on a plot of band number versus wavelength for the observed peaks.

The Matlab code used for performing the above processes is given in the Appendix.

## **5. GENERATION OF PRE-FLIGHT FLAT FIELDING COEFFICIENTS**

Prior to flight the DH1 sensor was flat fielded to eliminate spatial variation in radiometric response. All pre-flight flat fielding data was acquired in room 1036 of building 216, NRL. The instrumentation used is shown in Figure 12. The OL Series 455 integrating sphere (Optronic Laboratories, Inc.) provides a large area, uniform, diffusely radiating source with a near normal luminance. A 150-watt tungsten quartz-halogen lamp serves as the source and can be varied over several decades of luminance without changing the color temperature. Luminance adjustment is achieved using a precision micrometer-controlled variable aperture

between the lamp and the integrating sphere. A precision silicon detector-filter combination with an accurate photopic response (mounted in the integrating sphere wall) monitors the sphere's luminance. The source is controlled via a separate electronics display console and power supply (Optronic Laboratories, Inc.). The controller contains a highly regulated DC constant current power supply and has been designed for source stability and accuracy. The lamp current can be varied from zero to full power rating and is displayed to within .05% of the true lamp current via a digital display.



*Figure 12: Instrumentation used for Flat Fielding of the DH1 Sensor.*

By focusing the sensor onto the face of the integrating sphere and collecting image frames for various levels of luminance a model of each pixel's radiometric response was determined. The model was then used to determine a set of coefficients that could be used to correct for spatial variances in response. Generation of flat-fielding coefficients involved three steps: preparing the data, overlaying the integrating sphere source spectrum on the camera bands, and finding the gain and offset corrections for each pixel. The integrating sphere data was prepared by applying the re-sampling technique described in Section 4. This provided regularized flat-field data that was calibrated both spectrally and spatially. The spectral calibration

is essential for the flat-fielding procedure. Next the source spectrum provided by the manufacturer was interpolated onto the camera band center wavelengths determined in the spectral calibration. A cubic spline interpolator was used. The source spectrum was normalized so that the maximum intensity of the maximum integrating sphere luminance setting was unity. Note that since an aperture controlled the luminance with the lamp current held constant, it was assumed that the source spectrum did not change shape across the four or five different luminance settings. Finally a linear least squares fit was used to estimate a gain and offset correction for each pixel in the re-sampled image. Specifically, for each pixel a linear fit was made to a plot of pixel response versus normalized source intensity for the four or five luminance settings used. Note that since the source spectrum was normalized to unit maximum, this procedure provided only a flat-field correction. Although the data available would have made a full radiometric calibration possible, this was judged to be unnecessary since the radiometric response of the FPA, which was not temperature controlled, was not expected to be stable.

The Matlab code used for performing the above processes is given in the Appendix.

## **6. REAL-TIME IN-FLIGHT CORRECTIONS**

The correction and calibration procedures discussed in Sections 4 and 5 were implemented for real time operation in the DH1 data acquisition system. The corrections were included in the sensor controller software (Figure 5) which ran on the sensor controller computer. This software was implemented in C++ under the Windows NT operating system. To perform the corrections, a combined set of calibration coefficients was first generated that included both FPA re-sampling and flat-field corrections. A total of seven parameters were required for each pixel in the corrected image frames. The first two parameters were the x and y position on the focal plane to be mapped to the pixel. The next four parameters were the multiplicative coefficients for the four nearest neighbor FPA pixels for interpolation. These coefficients were the product of a bilinear interpolator for the pixel with the pixel gain correction. The last parameter was the pixel offset. Only the central fifty percent of the spatial extent of the camera was used. Although this was done to reduce system bandwidth requirements, little data was actually discarded since the image of the spectrometer slit did not fill the entire FPA. This down selection of the data was accomplished during the calibration and correction processing by using only the parameters for the central 128 pixels. The camera controller software loaded the correction parameters from a file at run time and applied them to each FPA frame as they were read from the frame grabber board.

## **7. GENERATION OF POST-FLIGHT FLAT FIELDING COEFFICIENTS**

After flying the DH1 sensor it was discovered that the radiometric response of the CCD was not correctly "flattened". It was found that the flat-fielding coefficients acquired during the pre-flight flat fielding procedure were insufficient. As a result, a method of post-flight flat fielding was developed. The two differences between the pre and post-flight approaches are in the calibration source used and in the mathematical modeling of the pixel radiometric responses. The source used for post-flight flat fielding is simply the reflectance spectra of water acquired during the field data collects, as opposed to the integrating sphere used for pre-flight flat fielding. Also, the modeling used for post-flight flat fielding is somewhat simpler. A two-point linear fit of the radiometric response data is used, as opposed to the five-point least squares fit used for pre-flight flat fielding. The two-point fit is necessary because only two intensity values are available. One, the intensity of the water reflectance during the course of the fly over and two, the intensity values associated with a dark frame collected while the camera bay door was closed. The simple linear fit provides an offset (the y-intercept of the linear fit) and a gain (the slope of the linear fit) for each of the FPA pixels.

## **8. RESULTS AND DISCUSSION**

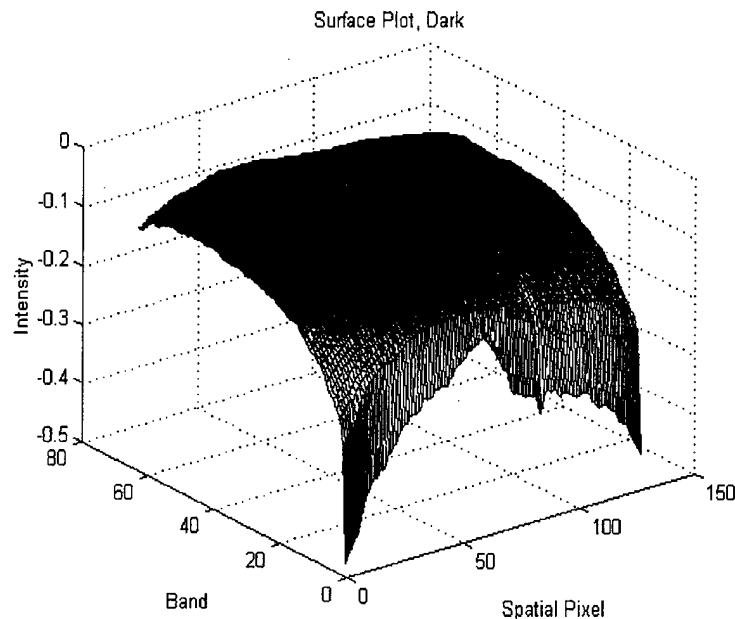
### **Spectral and Spatial Re-sampling and Calibration**

In general, spectral and spatial re-sampling of the FPA was found to be successful in reducing image distortion. The real time software correction for re-mapping of the collected imagery proved both useful and robust in practice. Some spectral and spatial misregistration is still present but this can be solved. There are two obvious areas for improvement in the correction and calibration procedures described (Section 4). The first is the peak finding phase of the generation of FPA re-sampling parameters. As was mentioned earlier, the peak finding procedure used was error prone and required significant manual intervention. One clear source of error is the residual non-uniformity in the data after dark frame subtraction. A possible remedy for this would be to apply a rough two-point flat-field correction to the calibration lamp data using the integrating sphere data with approximate band assignments. It may also be beneficial to further constrain the peak finding optimization step using knowledge about the structure of the calibration data. For example, it is known that the lamps have a fixed spectrum and that they appear as a spatial point source in the far field of the camera lens. One approach would be to try to fit each lamp image (each spatial location) to a fixed number of peaks lying on some smooth curve. Alternatively, one could attempt to fit the data to the known lamp spectrum warped onto a smooth curve.

The second area for improvement is in the interpolator used in the real time corrections. The bilinear interpolator used to date is simplistic and sub-optimal. There exist image interpolators that are optimal with respect to their distortion of the spatial spectrum of the image. Improvements to a first order (four nearest neighbors) interpolator could be realized, and if processor bandwidth permits, higher order interpolators could be considered.

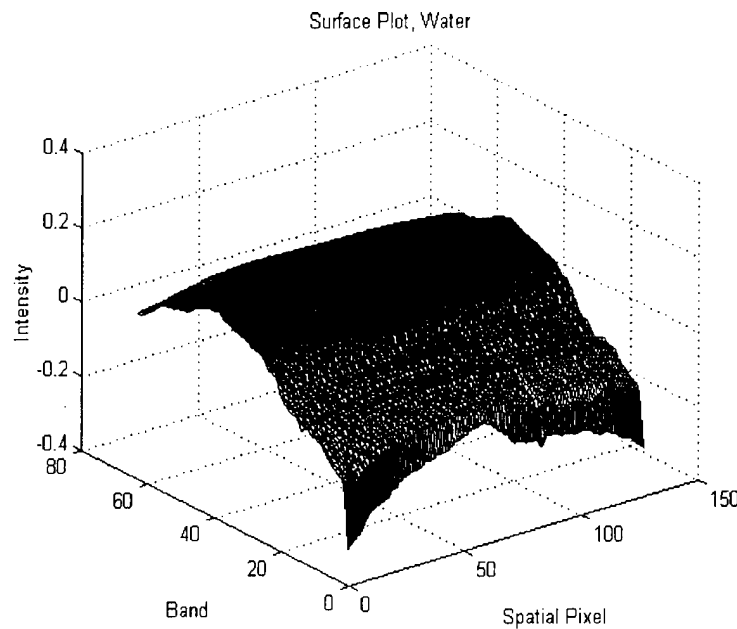
### **Pre-Flight Flat Fielding**

As previously stated, pre-flight flat fielding was found to be inadequate for field measurements. It was found that the radiometric response of the CCD still varied as a function of space even after the flat-fielding coefficients were applied to the data. This is clearly demonstrated in Figure 13 which is one frame of a series of dark frames collected during the March 5 flight test. The dark frames were collected when the camera bay door was closed and the sensor was in complete darkness. It is evident from Figure 13 that use of the pre-flight flat fielding coefficients was not successful in "flattening" the radiometric response of the FPA. If the array were flattened correctly all spatial and spectral pixel values would be of equal intensity (disregarding small random noise fluctuations). However, it is evident that this is not the case especially along the edges and corners of the array.



*Figure 13: Single Dark Frame Corrected using Pre-flight Flat Fielding Coefficients.*

Figure 14 also demonstrates that using the pre-flight flat fielding coefficients was not successful in "flattening" the radiometric response of the FPA. The single frame is from a series of frames collected while flying over water. In this case, if the array were flattened correctly each series of spectral pixels would be identical for each of the cross-track spatial pixels, with the series of spectral pixels representing the reflectance spectrum of water. Thus, the pixels would be equal in intensity along the spatial direction but not equal along the spectral direction. Again, this is clearly not the case, especially along the edges and corners of the array.

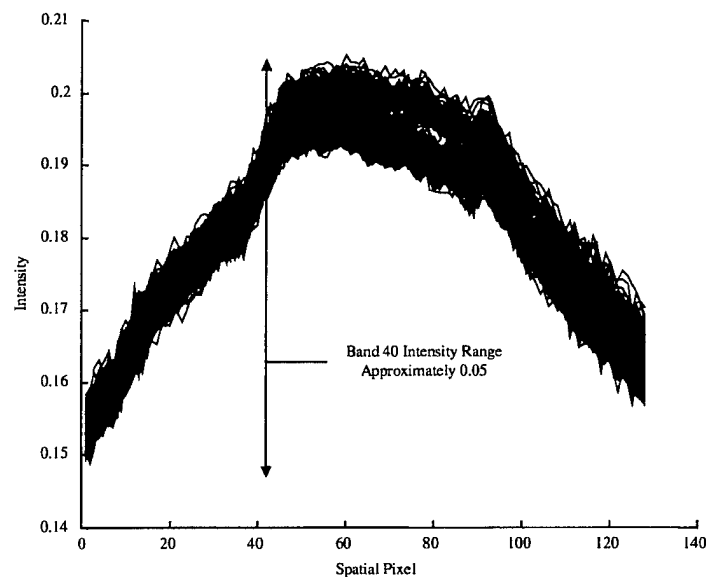


*Figure 14: Single Water Frame Corrected using Pre-flight Flat Fielding Coefficients.*

Figure 15 provides a more quantitative look at the spatial variations in radiometric response for the FPA (using the pre-flight flat fielding coefficients). The overlaid profiles represent fluctuations in intensity of spectral band 40 as a function of the 128 cross-track spatial pixels. The 200 profiles correspond to consecutive frames of data collected over water. Thus, we would expect the intensity of band 40 to be equal in value if the array were correctly "flattened". It is evident that this is not the case and that the intensity of band 40 varies significantly across the cross-track spatial direction of the array. In fact, with an expected

band 40 value of 0.2 for each of the 128 spatial pixels it is found that there is approximately a 25% change in response from the edges of the array (spatial pixels 1 and 128) to the middle of the array (pixel 60).

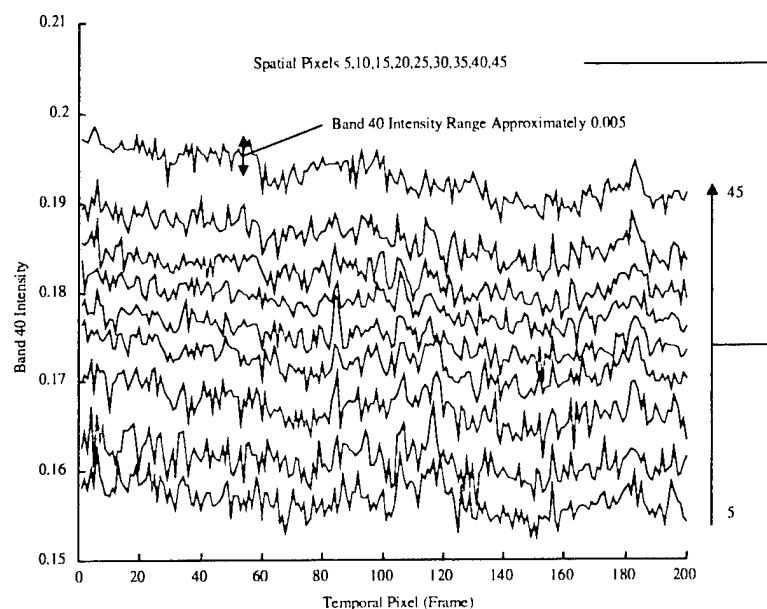
An explanation of why the pre-flight flat fielding is not sufficient is realized by examining the means by which the coefficients were acquired. Recall that the gain and offset were determined by applying a five-point least squares fit to a model of each pixel's radiometric response, with the slope of the fit corresponding to the pixel gain and the y-intercept of the fit corresponding to the pixel offset. This would provide a good measure of the gain and offset coefficients if the response of the CCD were linear over its entire dynamic range. Unfortunately this is not the case and, as a result, the response of the individual pixels is not modeled correctly for all luminance levels using a linear fit. The pixels may be modeled well for a given range of intensities but not over the entire dynamic range. This will result in pixels illuminated at intensity values outside of the properly modeled range to be incorrectly flat fielded using the calculated coefficients. Such behavior is evident in Figure 13 that shows the poorly flat fielded pixels along the edges and corners of the array. This makes sense because these regions correspond to areas of low optical throughput. Therefore, the low intensity light illuminating these pixels likely falls in regions of the sensors dynamic range that have been poorly modeled.



*Figure 15: Spatial Variations of Band 40 for 200 Water Frames.*



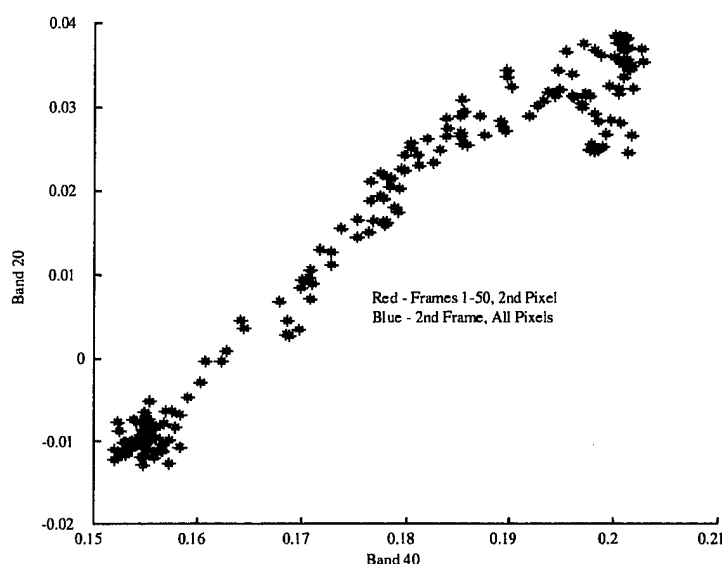
Figure 16 demonstrates that the radiometric response variations in the cross-track spatial direction are much larger than in the down-track temporal direction. In Figure 16 the same 200 frames of water data that were used in Figure 15 have been examined. However, in this case the profiles correspond to the temporal changes of band 40 for several of the 128 spatial pixels. The changes in band 40 have been monitored in the down-track direction and not the cross-track direction. From this it is evident that the variations in short term temporal response of the FPA are much smaller than the aforementioned variations in spatial response. For instance, if we examine band 40 of spatial pixel 45 for the FPA we find that there is only a 2.5% change in response over the course of 200 frames of collected data. This holds true for the other cross-track spatial pixels as well.



*Figure 16: Temporal Variations of Band 40 for 200 Water Frames.*

The phenomena demonstrated in Figure 15 and Figure 16 are related in Figure 17. This figure overlays the scatter plots of band 20 versus band 40 for all of the cross-track spatial pixels of a single water frame (blue) and the second cross-track spatial pixel of 50 consecutive water frames (red). Again, it is demonstrated that the intensity variations for a given band are much larger in the cross-track spatial direction than in the down-track temporal direction. It is also demonstrated in this figure that classification of similar pixels would be more difficult in the cross-track spatial direction than in the down-track temporal direction.

The tight clustering of the temporal pixels (red) and the relatively loose clustering of the spatial pixels (blue) clearly show this. The effects of poor flat fielding and the resulting errors in sample clustering are briefly revisited below, with an emphasis on how they effect the real-time anomaly detection algorithms.

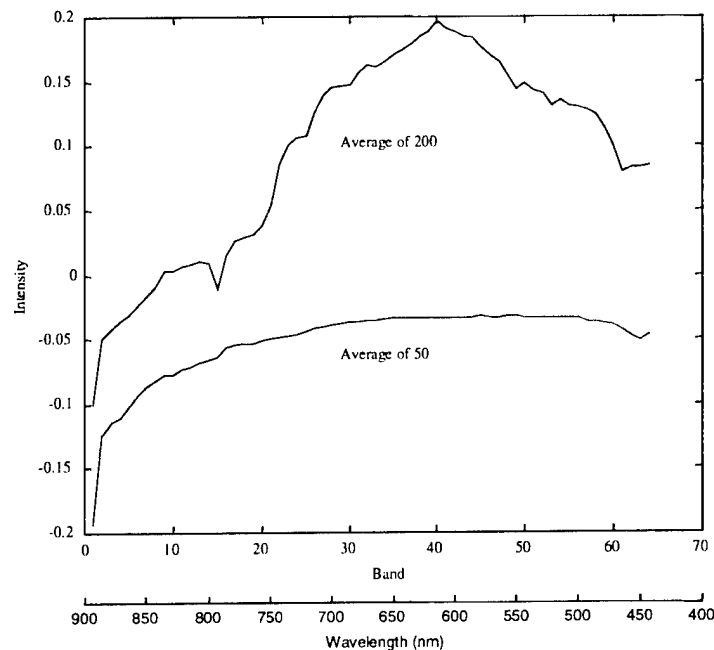


*Figure 17: Scatter Plot of Water Frames in the Spatial and Temporal Directions.*

Figure 17 also lends insight into problems associated with the pre-flight flat fielding. Note that the cross-track pixels (blue) only show large variations in the intensity direction (i.e. the two bands are moving together). This can also be seen in Figure 14. In both cases, the spectrum is simply moving up and down in intensity while the shape of the spectrum is remaining constant. From this it is evident that the dominant source of FPA pattern noise is just the spatial "falling off" of the CCD at the edges of the array. If spatial drift in CCD response was a large contributor to FPA pattern noise the cross-track pixels would vary in both intensity and spectral shape. This would result in the cluster of blue pixels being more elliptical in shape. In short, it appears that the poor results of the pre-flight flat fielding are largely the result of a deficiency in the linear calibration and not the result of spatial drift in CCD response over time.

## Post-flight Flat Fielding

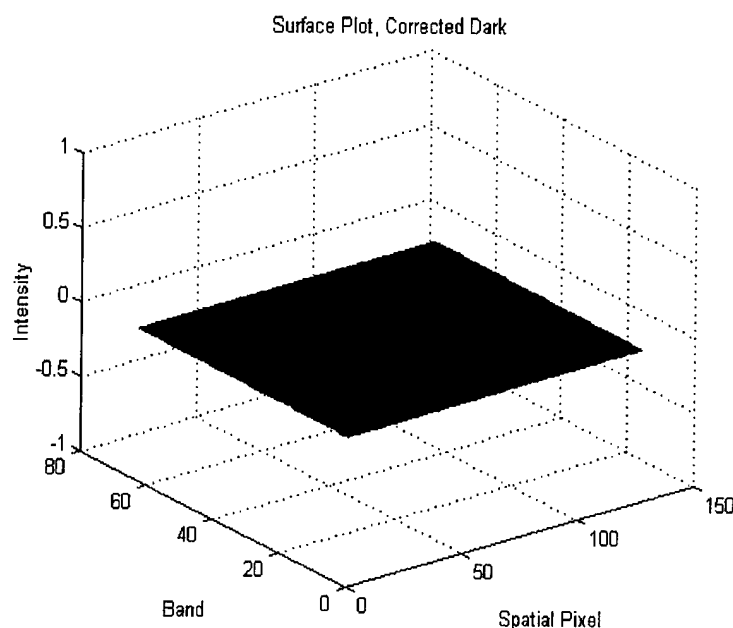
As previously mentioned, a method of post-flight flat fielding was developed (Section 7). Figure 18 shows two of the 256 spectra used for calculating the post-flight flat fielding coefficients. Two spectra were used for each of the 128 cross-track spatial pixels. The water reflectance spectrum (blue) and dark frame spectrum (red) are the average of 200 and 50 consecutive spectra, respectively. In this case, the spectra correspond to spatial pixel 60 of the FPA. The post-flight flat fielding is really a re-correction of the original pre-flight flat fielded data. Because spectral registration of the sensor was found to not change as a function of extended time the original spectral calibration of the data holds. It is only the flat fielding correction that must be re-done. A simple two-point linear fit was used to estimate a gain and offset correction for each pixel in the FPA. The slope of this fit was taken to be the pixel gain and the y-intercept was taken to be the pixel offset. Of course, in the case of this simple two-point fit, the offset is simply the intensity value of the dark spectrum.



*Figure 18: Water Reflectance Spectrum (Blue) and Dark Frame Spectrum (Red).*

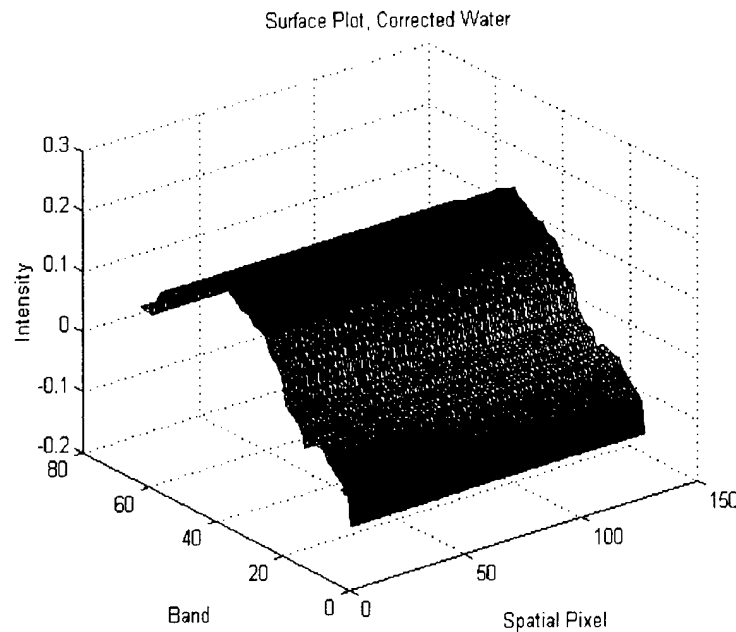
At first it may seem that the simpler two-point linear fit would be no more useful in flat fielding the sensor than the five-point least squares fit used for pre-flight flat fielding. However, it is important recall that the five-point fit modeled the individual pixel responses using intensities spread over the entire dynamic range of the sensor. This resulted in the response of the pixels being poorly modeled for lower luminance

levels. Unfortunately, to avoid having high intensity pixels saturate the detector, a large majority of the image pixels must be kept to low intensity values during the course of a data collect. Therefore, the pre-flight flat fielding is insufficient. In contrast the two-point fit focuses on modeling the pixel responses at lower luminance levels. The intensity of the water source spectrum is more typical of the image pixel intensities found during a collect. Therefore, a model of the pixels radiometric response is calculated, better gain and offset coefficients are acquired and thus the flat fielding is found to be better overall.



*Figure 19: Single Dark Frame Corrected using Post-flight Flat Fielding Coefficients.*

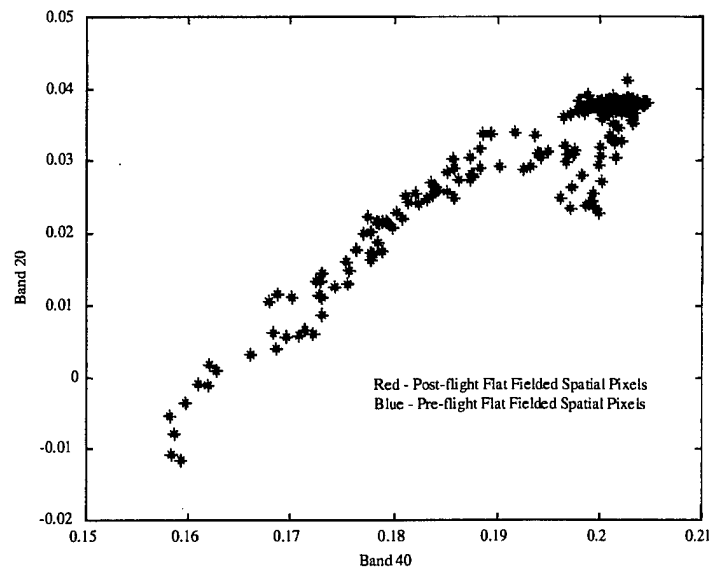
Figure 19 demonstrates that the flat field coefficients acquired from the post-flight flat fielding procedure are indeed useful. Figure 19 is a different dark frame than the one used in Figure 13. The dark frame is one collected during a later portion of the data collect. In this case, the post-flight flat fielding coefficients have been applied to the dark frame instead of the pre-flight coefficients. It is evident that the radiometric response of the FPA has been "flattened". As expected, all values across the FPA are equal in intensity. This is even true at the edges and the corners of the array; areas that showed considerable radiometric variances as a result of pre-flight flat fielding.



*Figure 20: Single Water Frame Corrected using Pre-flight Flat Fielding Coefficients.*

Figure 20 also demonstrates that using the post-flight flat fielding coefficients was successful in "flattening" the radiometric response of the FPA. The single water reflectance frame is a different water frame than the one used in Figure 14. The water frame is one collected at the end of the data collect (the other side of Blossom Point) instead of the beginning. Again, the post-flight flat fielding coefficients have been applied and the FPA has been successfully flattened, even along the edges and corners of the array. Note that each series of spectral pixels is identical for each of the cross-track spatial pixels, with the series of spectral pixels representing the reflectance spectrum of water.

Figure 21 further demonstrates that the post-flight flat fielding coefficients were successful in "flattening" the radiometric response of the FPA. Figure 21 is an overlay of the scatter plots of band 20 versus band 40 for all of the cross-track spatial pixels of a single water frame using the pre-flight flat fielding coefficients (blue) and the post-flight flat fielding coefficients (red). Note that the intensity variations for a given band are much larger using the pre-flight corrections. It is also demonstrated in this figure that classification of similar pixels would be more difficult using the pre-flight coefficients than the post-flight ones. This is shown through the tight clustering of the post-flight corrected pixels (red) and the relatively loose clustering of the pre-flight corrected pixels (blue). The effects of poor flat fielding and the resulting errors in sample clustering are briefly revisited below, with an emphasis on how they effect the real-time anomaly detection algorithms.



*Figure 21: Scatter Plot of Flat Fielded Water Frames in the Spatial Direction.*

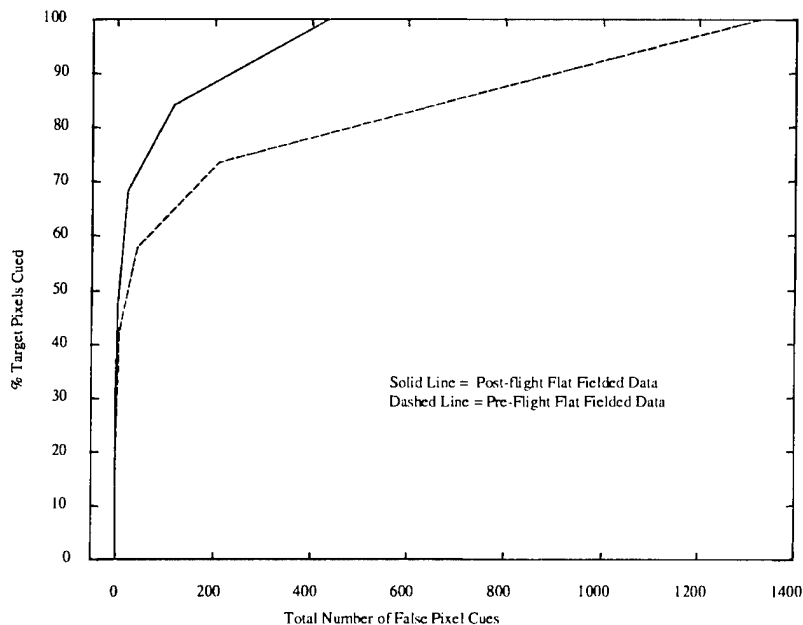
It should be noted that while the post-flight flat fielding does a much better job of "flattening" the image data than the pre-flight flat fielding that it will still show limitations. It is likely that pixels of very high intensity (near saturation) will still not be flattened correctly. In the case of examining the dark frame and the water frame (Figures 19 and 20) no high intensity pixels are present. However, it should also be noted that the occurrence of high intensity pixels is much less than that of low intensity ones. Therefore, the post-flight flat fielding is still a much better alternative.

### **Real-time Anomaly Detection**

The subspace R-X and LBG clustering algorithms were run on the pre-flight and post-flight flat fielded data. The algorithms were run in real-time with the same processor used for the in-flight data collections. The effects of poor flat fielding and the resulting errors in anomaly detection are briefly visited below. An emphasis has been placed on how flat fielding effects the real-time anomaly detection algorithms. In short, it is known that if the algorithms do not succeed in clustering spectrally similar pixels together (identify how target pixels are different from surrounding background pixels) an increase in false alarms will occur. As expected, it has been found that proper flat fielding of the data can help to reduce false alarms. This is particularly true for the R-X algorithm, which outperformed the LBG algorithm, at

least for the chosen scene of interest and chosen algorithm parameters. The performance increase can be attributed to proper flat fielding diminishing spectral differences associated with changes in spatial response of the FPA. As a result, the algorithm can "focus" on the true spectral differences between the targets and background regions of interest.

The parameters used by the two real-time algorithms follow. The parameters were not optimized for best performance but have been found to produce above average results. They have been largely chosen through trial and error during the in-flight measurements and through some preliminary post-examination of the spectral data cubes. For both algorithms spectral bands 40-54 were used, corresponding to the wavelength region from 725-840 nm. Also, both algorithms employed spatial filtering of the data. This was found to largely reduce the number of false alarms. The R-X algorithm retained principal components 4-10 and averaged 10 effective lines in its recursive filter. The LBG algorithm employed 10 clusters. During initialization the algorithm used 10 lines for cluster recognition, 1 line for decimation and went through 10 iterations. Two iterations were used during operation.

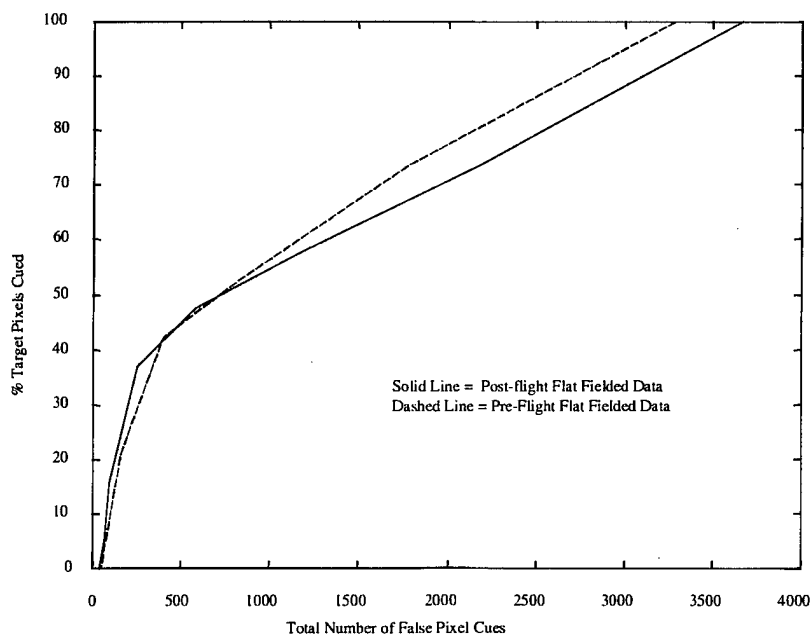


*Figure 22: Plot of Real-time R-X Algorithm Results for the Flat Fielded Data.*

Figure 22 is a plot of the percentage of target pixels cued versus the total number of false pixels cued, using the subspace R-X algorithm, for both the pre and post-flight flat fielded data. It is evident from

the superimposed profiles that the algorithm performs up to 67% better on the post flight flat fielded data than on the pre-flight flat fielded data. This is for the case of detecting 100% of the target pixels. It is also evident that the effect of flat fielding on algorithm performance decreases as a function of decreasing percentage of target pixels cued and is negligible below 40%.

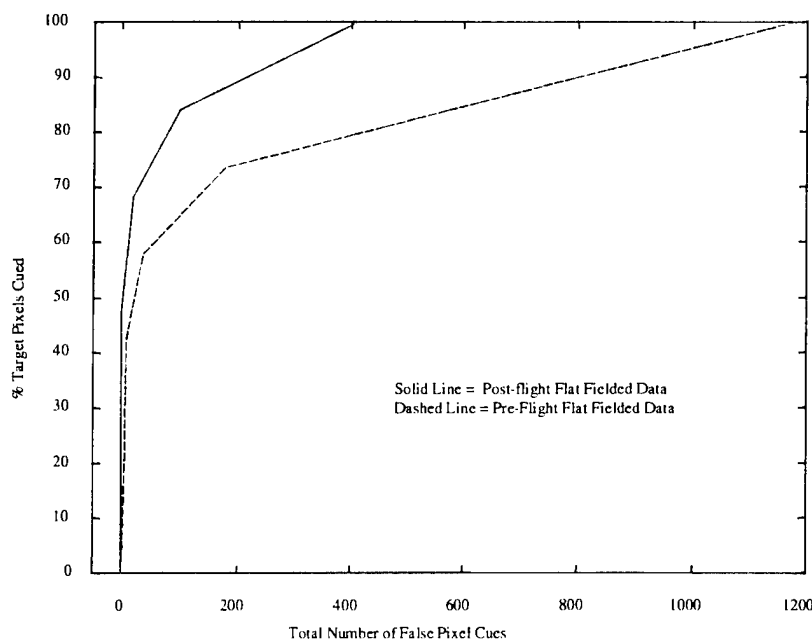
Figure 23 is a plot of the percentage of target pixels cued versus the total number of false pixels cued, using the LBG clustering algorithm, for both the pre and post-flight flat fielded data. Unlike the previous case, it is evident from the superimposed profiles that there is little positive effect on algorithm performance with respect to the pre and post-flight flat fielding of the data. In fact, for the post-flight flat fielded data the algorithm performs somewhat worse at higher percentages of target pixel cueing and only slightly better at lower percentages. However, it should also be noted that the LBG algorithm's performance is much worse (at least for this scene and chosen parameters) than that of the R-X algorithm. The R-X algorithm performs 60% better than on the pre-flight flat fielding data and 88% better on the post-flight flat fielding data. This assumes a comparison of the algorithms at 100% of the target pixels being cued. These results should be taken into consideration when considering the effects of proper flat fielding on algorithm performance.



*Figure 23: Plot of Real-time LBG Algorithm Results for the Flat Fielded Data.*



Lastly, Figure 24 is a plot of the percentage of target pixels cued versus the total number of false pixels cued, using both the subspace R-X algorithm and the LBG algorithm, for both the pre and post-flight flat fielded data. The combined algorithms performed virtually identical to the R-X algorithm alone, with the algorithms performing up to 66% better on the post flight flat fielded data than on the pre-flight flat fielded data. Again, this was for the case of detecting 100% of the target pixels. Using both algorithms in place of just the R-X algorithm resulted in a performance improvement of only 1-3%. This is expected due to the comparatively low performance of the LBG algorithm with respect to the R-X algorithm. Thus the LBG algorithm has little to contribute. However, it should again be noted that these comparisons are only based on the scene of interest and the chosen algorithm parameters. It is likely that the benefits of using the two algorithms in parallel will be better realized through optimization of the chosen parameters through a detailed post-examination of the collected spectral data cubes and further analysis of the algorithms on different scenes. It is also possible that the LBG algorithm will perform more strongly than the R-X algorithm for different scenes of interest. It may be found that running the algorithms in parallel will provide the potential for reducing false alarm counts for some scenes of interest but that running the algorithms in a one-or-the-other scenario may prove beneficial for other scenes of interest. This remains to be seen and will be examined in future studies.



*Figure 24: Plot of Real-time RX and LBG Algorithm Results for the Flat Fielded Data.*

## 9. REFERENCES

- [1] "Hyperspectral Reconnaissance Experiment at Visible Wavelengths with Real-Time Cueing and Detection", C.M. Stellman, F. Bucholtz, G.G. Hazel, D. Brown, G. Lynn and J.V. Michalowicz, Submitted to International Journal of Remote Sensing, August 1998.
- [2] "Multispectral Field Measurements for the Multispectral Overhead IR/EO Surveillance Program", W. Shaffer, A. Schaum, E. Ashton, M. HcHugh and J.V. Michalowicz, NRL Formal Report 5621-96-9831, 13 September 1996.
- [3] "Adaptive Multiple-Band CFAR Detection of an Optical Pattern with Unknown Spectral Distribution", I.S. Reed and X. Yu, IEEE Transaction on Acoustics, Speech and Signal Processing, Vol. 38, No. 10, October 1990.
- [4] "An Algorithm for Vector Quantizer Design", Y. Linde, A. Buzo and R. Gray, IEEE Transaction on Communication Theory, Vol. 17, June 1993.

## 10. APPENDIX

### *philsfindcoord.m*

*% SCRIPT TO DO AUTOMATED PEAK FINDING FOR DH1 REMAPPING.*

*% Input: fixcubelxxfxx.mat*

*% Output: temp2.mat*

clear

peakguess=[];save temp2 peakguess;clear peakguess;

*%LOAD IN BINNED DATA. APPLY 2 POINT CORRECTION*

dim = [512/8 512/2 16];

fstop = 40;

data1=zeros(dim(2),dim(1));

*% DEFINE WINDOWS*

*% 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24*

ystrt = [25 35 44 55 63 70 80 90 97 106 115 125 133 142 150 160 169 179 189 198 210];

windy = [20 20];

xstrt = [1 35 54];

windx = [20 20 10];

sz = size(file\_in);

back = zeros(2,5);

load fixcubel50f56

sz(1)=24;

for ii = 1:sz(1),

*%data = philsconvert(stripblank(file\_in(ii,:)), dim);*

*%data = squeeze(fixcube(:,:,sz(1) - ii + 1));*

data = squeeze(fixcube(:,:,ii));

data1 = data1+data;

if or(ystrt(ii)+windy(ii) < 128, ystrt(ii) > 128)

back(1,1) = floor(windy(ii)/2);

else

```

back(1,1) = 128 - ystrt(ii) + 1;
end
for jj = 1:size(xstrt,2),
    %SELECT SUBREGION
    back(1,2) = (floor(xstrt(jj)/8) + 1)*8 - xstrt(jj) + 1;
    back(1,3) = back(1,2) + 8;
    if back(1,3) >= windx(jj)
        back(1,3) = windx(jj)-1;
    end
    [xstrt(jj) ystrt(ii)]
    data2 = data(ystrt(ii):ystrt(ii)+windy(ii),xstrt(jj):xstrt(jj)+windx(jj));
    save temp1 data2;
    back(1,4) = data2(1,1);
    back(1,5) = data2(1,back(1,2)+1);
    back(2,1) = data2(1,windx(jj));
    back(2,2) = data2(windy(ii),1);
    back(2,3) = data2(windy(ii),back(1,2)+1);
    back(2,4) = data2(windy(ii),windx(jj));
    back(2,5) = 0;
    %CRUDELY IDENTIFY PEAKS BASED ON PEAK VALUE AT NEAREST COORDINATE
    init_guess = philscrudepeaks(data2,.005);
    init_guess = cat(1,init_guess,back)
    %   init_guess(:,6)   =   ones(size(init_guess,1),1)*(((data2(1,1)+   data2(1,windx(jj))   +
    data2(windy(ii),1) + data2(windy(ii),windx(jj))))/(4*size(init_guess,1)))
    %APPLY NONLINEAR LEAST SQUARES FIT TO EXTRACT PEAK FEATURES
    final_guess = leastsq('myfun',init_guess)
    final_guess = final_guess(1:(size(final_guess,1)-2),:);
    final_guess(:,1)=final_guess(:,1)+(ystrt(ii)-1);
    final_guess(:,2)=final_guess(:,2)+(xstrt(jj)-1);
    load temp2; peakguess = cat(1,peakguess,final_guess);
    save temp2 peakguess; clear peakguess

```

```

end
end
%FILTER OUT DEAD PEAKS (SMALL PEAKS WITH BIG VARIANCES
load temp2;
count = 0;xmaxwidth=2.0;ymaxwidth=4.0
for kk = 1:size(peakguess,1),
if and((peakguess(kk,4)<ymaxwidth),(peakguess(kk,5)<xmaxwidth)),
count=count+1;
peakguess1(count,:)=peakguess(kk,:);
end
end
figure(1);colormap(gray);imagesc(agc(data1));
figure(2);plot(peakguess1(:,2),peakguess1(:,1),*');axis([1 64 1 256]);set(gca,'YDir','Reverse');
figure(3);imagesc(agc(data1));colormap(gray);hold
on;plot(peakguess(:,2),peakguess(:,1),r*');axis([1 64 1 256]);set(gca,'YDir','Reverse');
figure;imagesc(agc(data1));colormap(gray);hold
on;plot(peakguess1(:,2),peakguess1(:,1),r*');axis([1 64 1 256]);set(gca,'YDir','Reverse');

```

### ***spatialregister.m***

*% SCRIPT TO RUN AFTER PHILSFINDCOORD.M*

*% DO SPATIAL REGISTERING*

*% Input: culled.mat (Hand edited peaks from temp2.mat)*

*% Output: stnd\_y.mat*

load culled

figure

plot(peakguess6(:,2),peakguess6(:,1),\*');axis([1 64 1 256]);set(gca,'YDir','Reverse');

hold on

xxx = 1:64;

for row = 1:21

```

line = polyfit(peakguess6(((row-1)*8 + 1):((row-1)*8 + 8),2),peakguess6(((row-1)*8 + 1):((row-1)*8 + 8),1),1);
plot(xxx,xxx*line(1) + line(2),'r')
mid(row) = 32.5*line(1) + line(2);
end
pos = 0:3:60;
%pos = [0 3 6 9 12 18 21 24 30 33 36 42 45 48 51 54 66 69];
line = polyfit(pos,mid,1);
std_Y = line(1)*pos + line(2);
figure
plot(pos,pos*line(1)+line(2))
hold on
plot(pos,mid,'r*')
save std_y std_Y

```

### ***phillsfindmap.m***

*% SCRIPT TO RUN AFTER SPATIALREGISTER.M*

*% Use identified peaks to regularize image and interpolate 'delta' vector map to all pixels*

*% Input: culled.mat (Hand edited peaks from temp2.mat)*

*% std\_y.mat*

*% Output: corcoeflxxfxx.mat*

clear

load culled

peakguess1 = peakguess6;

clear peakguess6

%indlog = ((peakguess1(:,1)<110)&(peakguess1(:,1)>85));

%std\_X = peakguess1(indlog,2);

%indlog = ((peakguess1(:,2)<15)&(peakguess1(:,2)>10));

%std\_Y = peakguess1(indlog,1);

%clear indlog;

```

stnd_X = peakguess1(81:88,2);
%stnd_Y = peakguess1([3 9 15 20],1);
load stnd_y
count = 0;
for ii = 1:length(stnd_Y),
for jj = 1:length(stnd_X),
count=count+1;
goodyx(count,:) = [stnd_Y(ii) stnd_X(jj)];
end
end
badyx = peakguess1(:,1:2);
clear count ii jj kk maxwidth peak*
figure;
subplot(2,1,1);plot(badyx(:,2),badyx(:,1),*',goodyx(:,2),goodyx(:,1),'o');set(gca,'YDir','Reverse');axis([1 64 1 256])
%goodveto = [0 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1]';
%badveto = [ 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1]';
%goodveto = [1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1]';
%badveto = [1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1]';
%goodveto=logical(goodveto);
%badveto =logical(badveto);
%goodyx = goodyx(goodveto,:);
%badyx = badyx(badveto,:);
%clear *veto;
subplot(2,1,2);plot(badyx(:,2),badyx(:,1),*',goodyx(:,2),goodyx(:,1),'o');set(gca,'YDir','Reverse');axis([1 64 1 256])
change = goodyx-badyx;
figure;
subplot(2,1,1);plot(badyx(:,2),badyx(:,1),*',goodyx(:,2),goodyx(:,1),'o');set(gca,'YDir','Reverse');axis([1 64 1 256])

```

```
subplot(2,1,2);quiver(badyx(:,2),badyx(:,1),change(:,2),change(:,1));set(gca,'YDir','Reverse');axis([
1 64 1 256])
```

```
% HERE'S THE ACTUAL INTERPOLATION STEP
```

```
XI = 1:64; YI = (1:256)';
```

```
chngy = griddata(badyx(:,2),badyx(:,1),change(:,1),XI,YI,'v4');
```

```
chngx = griddata(badyx(:,2),badyx(:,1),change(:,2),XI,YI,'v4');
```

```
% HERE'S WHERE WE REINTERPRET THE 'DELTA' VECTORS (CHNGX, CHNGY) AS A
PIXELWISE COORDINATE TRANSFORMATION
```

```
[X_perf,Y_perf] = meshgrid(1:64,1:256);
```

```
X_intrp = X_perf - chngx;
```

```
Y_intrp = Y_perf - chngy;
```

```
Y_intrp(Y_intrp>256) = 256; Y_intrp(Y_intrp<1) = 1;
```

```
X_intrp(X_intrp>64) = 64; X_intrp(X_intrp<1) = 1;
```

```
save corcoefl50f56 X_intrp Y_intrp X_perf Y_perf
```

```
clear lambda freq polycoefs bands stnd_* XI YI X_* Y_* badyx goodyx chngx chngy change
```

```
clear freqs fdelta S bandsp bandsm
```

*justwarp.m*

```
% FUNCTION TO APPLY RESAMPLING TO DHI DATA
```

```
function [warpd] = justwarp(data,coefile)
```

```
% Inputs: coefficient file name (eg. corcoeflxxfxx.mat).
```

```
% Output: resampled data cube.
```

```
dim = size(data);
```

```
eval(['load ',coefile]);
```

```
warpd = zeros(dim);
```

```
for ii = 1:dim(2),
```

```
temp0 = squeeze(data(:,ii,:));
```

```
temp1 = interp2(X_perf,Y_perf,temp0,X_intrp,Y_intrp,'linear');
```

```
temp2 = reshape(temp1,[dim(1) 1 dim(3)]);
```

```
warpd(:,ii,:) = temp2;
```



ii  
end

### ***BandAssign.m***

*% SCRIPT TO COMPUTE BAND ASSIGNMENTS*

*% Input: Hand enter bands and wavelengths below*

*% Output: bandslxxfxx.mat*

lambda = [850.89 819.01 785.48 760.15 669.92 645.63 587.09 435.14];

pixels = [ 3    7.5  12    15.8  27.5  31.0  40.1  60.2];

[polycoefs S] = polyfit(pixels,lambda,1); %Fit to linear dispersion model

[bands delta] = polyval(polycoefs,[1:64],S);

figure;

subplot(2,1,1);plot(bands);hold on;    plot(pixels,lambda,'go');plot(bands+delta,'r');plot(bands-  
delta,'r')

xlabel('Pixel');ylabel('Wavelength (nm)'); title('Band Assignments Krypton 75mm lens, fstop 8.0  
10/13/97')

text(40,800,['Band Spacing = ' num2str(abs(mean(diff(bands)))) ' (nm)']);

subplot(2,1,2);plot(lambda,lambda-polyval(polycoefs,pixels),'b\*');hold on;  
plot(bands,delta,'r');plot(bands,-delta,'r')

xlabel('Wavelength (nm)'); ylabel('Error in Fit (nm)'); title('Band Assignment Error')

save bandsl50f56 bands

### ***calibrate.m***

*% SCRIPT TO DO FLAT-FIELD CORRECTION.*

*% Input: integrating sphere data \*.il6*

*% corcoeflxxfxx.mat*

*% lampdat.mat*

*% bandslxxfxx.mat*

*% Output: gainoffl50f56.mat*

```

dim = [512/8 512/2 16];
file_in = [...
'd:\users\ghazel\data\darkhorse\feb98_ca\l50f56\blackbody\5056a.i16',
'd:\users\ghazel\data\darkhorse\feb98_ca\l50f56\blackbody\5056b.i16',
'd:\users\ghazel\data\darkhorse\feb98_ca\l50f56\blackbody\5056c.i16',
'd:\users\ghazel\data\darkhorse\feb98_ca\l50f56\blackbody\5056d.i16',
'd:\users\ghazel\data\darkhorse\feb98_ca\l50f56\blackbody\5056e.i16',
];
file_in = flipud(file_in);
data = [];
for ii = 1:5
datain = philsread(file_in(ii,:), dim, 0);
% if ii==5
% datain=datain(:,[2:2:16],:);
% end
data = cat(2,data,datain);
end
clear datain ii
dim = size(data);
load corcoefl50f56
warpd = zeros(dim);
for ii = 1:dim(2),
temp0 = squeeze(data(:,ii,:));
temp1 = interp2(X_perf,Y_perf,temp0,X_intrp,Y_intrp,'linear');
temp2 = reshape(temp1,[dim(1) 1 dim(3)]);
warpd(:,ii,:) = temp2;
ii
end
clear temp0 temp1 temp2
load lampdat
load bandsl50f56

```

```

lamponbands = spline(lampdat(:,1),lampdat(:,2),bands);
lamponbands = lamponbands/max(lamponbands);
intensity    =    [(100/1100)*ones(1,16)    (350/1100)*ones(1,16)    (600/1100)*ones(1,16)
(850/1100)*ones(1,16) ones(1,16)];
for x = 1:64
for y = 1:256
ydat = lamponbands(x)*intensity;
xdat = warpd(y, :, x);
gainoff(y,x,:) = polyfit(xdat,ydat,1);
end
x
end
clear x y xdat ydat
for f = 1:33
temp0 = squeeze(warpd(:,f,:));
temp0 = temp0.*gainoff(:, :, 1);
fixed(:,f,:) = (temp0 + gainoff(:, :, 2));
f
end
figure; imagesc(squeeze(fixed(65:192,1,:)));colormap(gray)
figure;
subplot(2,1,1)
plot(bands,squeeze(sum(sum(warpd(50:125,17:32,:),1),2)));xlabel('Wavelength (nm)');
title('Average Spectrum Before Gain and Offset')
subplot(2,1,2)
plot(bands,squeeze(sum(sum(fixed(50:125,17:32,:),1),2)));xlabel('Wavelength (nm)'); title('After
Gain and Offset with Lamp Data')
hold on
plot(bands,lamponbands*max(squeeze(sum(sum(fixed(50:125,17:32,:),1),2))), 'go')
figure
subplot(2,1,1)

```

```

plot(
squeeze(sum(sum(warpd(:,17:32,35:60),3),2))/max(squeeze(sum(sum(warpd(:,17:32,35:60),3),2))))
;xlabel('Spatial Pixel'); title('Spatial Profile Before Gain and Offset');axis([0 256 0.5 1])
subplot(2,1,2)
plot(
squeeze(sum(sum(fixed(:,17:32,35:60),3),2))/max(squeeze(sum(sum(fixed(:,17:32,35:60),3),2))))x
label('Spatial Pixel'); title('Spatial Profile After Gain and Offset');axis([0 256 0.5 1])
save gainoffl50f56 gainoff
%clear gainoff fixed temp0 warpd data intensity lampdat lamponbands X_* Y_* dim f ii
%clear bands file_in
geoff_coef.m
% SCRIPT TO GENERATE COMBINED CORRECTION/CALIBRATION COEFFICIENTS
READABLE BY PC1 SOFTWARE

% Input: gainofflxxfxx.mat
% corcoeflxxfxx
% Output: coefslxxfxx.cor
%clear
fid = fopen('d:\users\ghaze\code\matlab\darkhorse\schuler\coefsl50f56.cor','w');
%load calibration_binned
%offsets = squeeze(cal_corr(:, :, 2)) ./ squeeze(cal_corr(:, :, 1));
%gains= squeeze(cal_corr(:, :, 1));
%offsets = offsets';
load gainoffl50f56
offsets = squeeze(gainoff(65:192, :, 2));
gains = squeeze(gainoff(:, :, 1));
fwrite(fid,offsets,'float32');
%clear
load corcoefl50f56
sz = size(X_intrp);
coef = zeros([sz 6]);

```

```

% FIND INDEXED COORDINATE
coef(:, :, 1) = floor(X_intrp); % X coord index...
coef(:, :, 2) = floor(Y_intrp); % Y coord index...
subX = X_intrp-floor(X_intrp);
subY = Y_intrp-floor(Y_intrp);
% FIND WEIGHTING COEFFICIENTS BASED ON SUB-PIXEL POSITION
coef(:, :, 3) = (ones(sz)-subX).*(ones(sz)-subY); % A coefficient...
coef(:, :, 4) = (subX).*(ones(sz)-subY); % B coefficient...
coef(:, :, 5) = (ones(sz)-subX).*(subY); % C coefficient...
coef(:, :, 6) = (subX).*(subY); % D coefficient...
% DEAL WITH BOUNDARIES BY BRUTE FORCE
for y = 1:sz(1)
for x = 1:sz(2)
if ((coef(y,x,1) == 0) & (coef(y,x,2) == 0)) %Just use corner
coef(y,x,1) = 1;
coef(y,x,2) = 1;
coef(y,x,3) = 1;
coef(y,x,4) = 0;
coef(y,x,5) = 0;
coef(y,x,6) = 0;
elseif (coef(y,x,1) == 0) %Just use two on the left edge
coef(y,x,1) = 1;
coef(y,x,3) = coef(y,x,3) + coef(y,x,4);
coef(y,x,5) = coef(y,x,5) + coef(y,x,6);
coef(y,x,4) = 0;
coef(y,x,6) = 0;
elseif (coef(y,x,2) == 0) %Just use two on the top edge
coef(y,x,2) = 1;
coef(y,x,3) = coef(y,x,3) + coef(y,x,5);
coef(y,x,4) = coef(y,x,4) + coef(y,x,6);
coef(y,x,5) = 0;

```

```

coef(y,x,6) = 0;
elseif ( (coef(y,x,1) == sz(2)) & (coef(y,x,2) == sz(1)) ) %Just use corner
coef(y,x,1) = sz(2)-1;
coef(y,x,2) = sz(1)-1;
coef(y,x,3) = 0;
coef(y,x,4) = 0;
coef(y,x,5) = 0;
coef(y,x,6) = 1;
elseif (coef(y,x,1) == sz(2)) %Just use two on right edge
coef(y,x,1) = sz(2)-1;
coef(y,x,4) = coef(y,x,3) + coef(y,x,4);
coef(y,x,6) = coef(y,x,5) + coef(y,x,6);
coef(y,x,3) = 0;
coef(y,x,5) = 0;
elseif (coef(y,x,2) == sz(1)) %Just use two on bottom edge
coef(y,x,2) = sz(1)-1;
coef(y,x,5) = coef(y,x,3) + coef(y,x,5);
coef(y,x,6) = coef(y,x,4) + coef(y,x,6);
coef(y,x,3) = 0;
coef(y,x,4) = 0;
end
end
end
%NOW FOLD IN THE GAINS
%for y = 1:sz(1)
%for x = 1:sz(2)
%coef(y,x,3)=coef(y,x,3)*gains(coef(y,x,2),coef(y,x,1));
%coef(y,x,4)=coef(y,x,4)*gains(coef(y,x,2),coef(y,x,1)+1);
%coef(y,x,5)=coef(y,x,5)*gains(coef(y,x,2)+1,coef(y,x,1));
%coef(y,x,6)=coef(y,x,6)*gains(coef(y,x,2)+1,coef(y,x,1)+1);
%end

```

```

%end
for ii = 3:6
coef(:, :, ii) = squeeze(coef(:, :, ii)).*gains;
end
% Zero Based arrays in C++
coef(:, :, 1) = coef(:, :, 1)-1;
coef(:, :, 2) = coef(:, :, 2)-1;
top = floor(sz(1)/2)-64;
for y = 1:128
for x = 1:sz(2)
fwrite(fid,coef(top+y,x,1:2),'uchar');
fwrite(fid,coef(top+y,x,3:6),'float32');
end
end
fclose(fid);

```